FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Publicidade baseada em Informação de Contexto

José Pedro Vieira Cardoso

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Miguel Pimenta Monteiro (Professor Doutor)

Responsável de acompanhamento: Telma Mota (Eng^{a})

20 de Julho de 2012

Publicidade baseada em Informação de Contexto

José Pedro Vieira Cardoso

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor Rui Filipe Maranhão de Abreu, Prof. Auxiliar da FEUP

Arguente: Doutor Luís Manuel Borges Gouveia, Prof. Associado da Universidade Fernando Pessoa

Orientador: Doutor António Miguel Pontes Pimenta Monteiro, Prof. Auxiliar da FEUP

---

20 de Julho de 2012

Resumo

Hoje em dia, o ambiente que no rodeia está sobrecarregado de informação, sendo bastante difícil para as pessoas conseguirem absorvê-la de forma eficiente. Um exemplo bastante comum, é a publicidade. Esta surge na forma de anúncios com texto, vídeos, imagens e sons, e surge como uma tentativa das empresas orientarem-nos para produtos e eventos que nos poderão interessar. Contudo, estes continuam a existir em demasia e, apesar de conseguirmos filtrar parte da informação que nos chega, uma pessoa não consegue filtrar tudo, o que leva a que sejam perdidas informações importantes devido a outras que não o sejam.

Neste contexto surgiu o conceito de *Context as a Service* (CaaS), que pretende explorar as oportunidades de uma plataforma que consiga recolher, armazenar e mediar o acesso de informações de contexto, de diversas fontes para um maior número de consumidores. Estas plataformas têm sido utilizadas com o objetivo de criar serviços orientados não apenas aos eventos, mas também aos utilizadores. O maior problema aqui continua a ser a maneira como são aproveitadas as informações de contexto, e de que maneira elas são usadas para auxiliar os serviços.

Esta tese de mestrado tem como objetivo a concepção da arquitetura dum sistema, com um *webservice* integrado. Este *webservice* será capaz de receber pedidos de anúncios vindos do exterior, processá-los e devolver resultados, devido a um algoritmo integrado, capaz de fazer classificação de anúncios, emparelhando-os com programas (*matching*), com base em diversas informações de contexto, nomeadamente ambiente do utilizador (localização, programa a ser visualizado), preferências dos utilizadores (programas preferidos) e preferências dos anunciantes (horários de emissão, público-alvo).

Neste relatório, é apresentada uma revisão bibliográfica ao nível de algoritmos de classificação e métricas usadas para cálculos de semelhanças, extração de regras de associação para aprendizagem de "tendências"dos anunciantes, e ainda é feita uma análise a serviços de distribuição de publicidade contextualizada.

Também são descritos os tipos de dados (programas e anúncios), a metodologia dividida por diferentes fases (pré-processamento, *matching offline* e *matching online*), devido aos diferentes tipos de critérios (estáticos e dinâmicos) a ter em conta durante o processo de seleção.

É apresentada a arquitetura física do sistema, com todos as partes distribuídos pela rede, assim como a arquitetura lógica, vertical e horizontal, com as diferentes camadas e componentes que constituem o sistema, respetivamente. É ainda apresentado o esquema relacional da base de dados, responsável pelo armazenamento destes.

Por forma a validar a solução implementada, foram executados alguns testes capazes de avaliar o desempenho do sistema, nas diferentes componentes. A análise comparativa entre serviços existentes e o serviço desenvolvido mostrou que existem pontos em comum pois, apesar de tudo, tratam-se de serviços de distribuição de publicidade com os mesmos objetivos, contudo também existem pontos em que diferem, pois são serviços executados em ambientes diferentes, nomeadamente Internet e *MEO*.

No final são apresentadas as conclusões tiradas ao longo da realização desta dissertação, os benefícios retirados deste trabalho, assim como é feita uma análise ao trabalho futuro que poderá vir a ser realizado com este sistema, nomeadamente melhoramentos e novas funcionalidades.

Abstract

Nowadays, the environment that surrounds us is overloaded with information, making it difficult for humans to absorb all of it, efficiently. One common example is advertising. It exists in the form of texts, videos, images, sounds, and appears as an opportunity for enterprises to guide us to product or events that may be of interest to us. However, there are still too many and, despite how capacity to filter information, the human being cannot filter everything, which leads to the loss of important information due to others that aren't.

In this context, came the concept Context as a Service (CaaS), which intends to explore the opportunities created by a platform that can collect, store and mediate access from diverse sources of information to an even greater number of consumers. This platforms have been used with the purpose to offer services, not only event-oriented, but also useroriented. The major problem here still continues to be how the context information is exploited, and how they can be used to leverage services.

The objective of this Master's thesis is conceiving an architecture for a system, with a built-in webservice. This webservice will be able to receive requests for ads comming from the outside, process them and return results, thanks to a built-in algorithm, capable of classifying ads, matching them with programs, based on context information, namely the user environmente (location, show being watched), user's preferences (favorite shows) and advertiser's preferences (schedules, target audience).

In this report, it's presented a state-of-the-art revision regarding classification algorithms and proximity measures for similarity calculation, association rule mining for learning advertiser's "trends", and also an analysis to other contextual advertising services.

Also, the different kinds of data (shows and ads) are described, the methodology to follow id divided by phases (pre-processing, offline matching and online matching), because of the different types of criterias (static and dynamic) to take into account during the selection process.

It's presented the system's physical architecture, with all the parts distributed over the network, as well as the logical architecture, vertical and horizontal, with all the system's different layers and components. Also, it's presented the relation schema of the database used to storage date.

In order to validate the implemented solution, some testes were performed capable of evalutating the system's performance, in all the different components. The comparative analysis between existing services and the one developed showed they have a lot in common because, afterall, their contextual adverting services providers and have the same objectives, however there are also some things they don't have in common because they don't run in the same environment, namely Internet and MEO.

In the end, some conclusions drawn during this dissertation are presenteed as well as an analysis to future work that could be done with this system, namely improvements and new functionalities.

Agradecimentos

Em primeiro lugar gostaria de agradecer ao Prof. Miguel Pimenta Monteiro e à Engª Telma Mota por toda a orientação, conselhos e tempo concedido durante o projeto, por estarem sempre dispostos a ajudar.

Em segundo lugar, queria agradecer à minha família por todo o apoio incondicional e respeito pelas minhas decisões que me deram ao longo destes cinco anos, principalmente nestes últimos 6 meses, durante o período de realização desta dissertação.

Em terceiro lugar, gostava de agradecer aos meus amigos mais chegados, não só pela ajuda, conselhos e bons momentos passados ao longo destes 6 meses mas também pelos 5 anos fantásticos nesta faculdade. Um grande obrigado à Catarina, Filipa, Sara, Marta, Sofia, Sónia, João, Pedro, Ricardo, Ricardo e Bruno.

Por último, gostava de agradecer aos restantes colegas de trabalho da FEUP e Instituto de Telecomunicações da Universidade de Aveiro que de alguma forma contribuíram para o sucesso deste trabalho, e professores pela disponibilidade e ensinamentos passados.

José Pedro Vieira Cardoso

"It's going to be legend...WAIT FOR IT ...DARY!"

Barney Stinson

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

API	Application Programming Interface
CaaS	Context as a Service
EPG	Electronic Programming Guide
FP-Growth	Frequent Pattern Growth
K-NN	K-Nearest Neighbor
JDBC	Java Database Connection
PT Inovação	Portugal Telecom Inovação
RGB	Red, Gree, Blue
SQL	Structured Query Language
SSH	Secure Shell

Capítulo 1

Introdução

Nos dias de hoje, a quantidade de informação existente é enorme e todos os dias cresce ainda mais, chegando mesmo a tornar-se imensurável. Contudo, as pessoas, de forma intuitiva, vão conseguindo contextualizar a informação que lhes chega e "filtrar" apenas aquela que acham importante ou interessante. As empresas publicitárias por sua vez tentam usar a informação de que dispõem para melhor construírem e distribuírem os anúncios, contudo estes continuam a existir de forma indiscriminada e o problema da sobrecarga de informação mantém-se.

Os anúncios publicitários devem ter como principal objetivo cativar a atenção das pessoas e fazer crescer dentro delas a vontade de querer comprar algo, sendo que o contexto em que surgem é de uma importância fulcral, pois só lhes será dada a atenção devida em situações favoráveis e nunca quando, por exemplo, a pessoa está impossibilitada de consumir essa informação devido a limitações humanas e/ou temporais.

1.1 Enquadramento

Face às necessidades de gerir o ambiente em que nos encontramos, surgiu o conceito *Context as a Service* (CaaS) [PRB+08], que pretende explorar as oportunidades de uma plataforma que consiga recolher, armazenar e mediar o acesso de diversas fontes para um maior número de consumidores. Com estas plataformas pretende-se que aplicações desenvolvidas, à volta de informações de contexto, consigam adaptar o seu comportamento, melhorando assim a experiência de utilização de serviços.

Os algoritmos de classificação também assumem aqui um papel importante, com capacidades dinâmicas, capazes de classificar conteúdos utilizando para isso métricas previamente definidas, ou seja, analisar as variáveis de ambiente em que uma pessoa se insere, em determinado momento, e encontrar o melhor anúncio possível para a mesma, certificando-se que o anúncio não passará despercebido. Este tipo de abordagem é utilizado por sistemas de recomendação [AT05], em que estes lidam com variáveis que definem um contexto, apresentando uma seleção de resultados

ao utilizador para que ele possa escolher o conteúdo a ser visualizado de acordo com as suas preferências.

Neste contexto, surgiu este projeto que consiste em desenvolver um algoritmo de classificação de conteúdo, baseado nos vários tipos de informações de contexto (espaciais e temporais), assim como preferências de utilizador, capaz de personalizar para cada utilizador serviços já existentes como, por exemplo, a *MEO Box*.

1.2 Problema

O problema com que nos defrontamos é então o de haver "sobrecarga de informação", isto é, existe demasiada informação, dispersa por demasiadas fontes, o que requere às pessoas que percam tempo, considerado valioso, na procura daquilo que mais lhes interessa. A informação em excesso faz com que informações relevantes passem por nós, sem que nós tenhamos noção daquilo que nos chega.

Para além destes problemas, o facto do ambiente que nos rodeia estar em constante mudança e haver demasiadas coisas a acontecer, torna-se complicado determinar em que situações está a pessoa pronta para lidar com determinados conteúdos, ou seja, de entre os vários tipos de conteúdos que existem (vídeos, imagens, textos, etc.) conseguir determinar pelo contexto qual o melhor a ser apresentado.

1.3 Instituição

A Portugal Telecom Inovação, SA (PT Inovação) é a empresa responsável pelo projeto. Foi constituída em 21 de Maio de 1999, sendo uma Sociedade Anónima, tem como atividade principal garantir o processo de inovação da Portugal Telecom, SA e empresas participadas (Grupo Portugal Telecom) através da prestação de serviços às mesmas. Tem a sua sede social em Aveiro. [kn:12e]

A principal missão da PT Inovação é promover o processo de inovação ao nível dos serviços, tecnologias e operações. O projeto desta dissertação será desenvolvido nas instalações do Instituto de Telecomunicações de Aveiro, localizado no campus da Universidade de Aveiro, em parceria com a PT Inovação.

1.4 Motivação

O domínio dos dados é extremamente volátil, pois todos os dias são produzidos novos conteúdos para substituir antigos, e existem ainda bastantes desafios ao desenvolver um algoritmo de classificação que consiga satisfazer os requisitos necessários, tais como a escalabilidade dos dados, as preferências do utilizador (programas preferidos) e dos anunciantes (público-alvo, horários de emissão, número de visualizações), o contexto em que o utilizador está inserido (programa a ser visto no momento, localização), os pesos a dar a cada variável usada na classificação, capacidade de resposta do algoritmo e capacidade de integração e expansibilidade. Todos estes fatores fazem

com que a decisão do anúncio a apresentar não seja algo trivial, mas bastante complexo devido ao elevado número e tipo de variáveis a ter em conta.

O maior benefício deste trabalho resulta no facto da emissão de anúncios, para além de orientado a eventos (novos produtos e promoções), seria também orientado ao utilizador, adaptando-se a este e permitindo uma experiência de utilização personalizável e, consequentemente, mais satisfatória e agradável.

Também o facto de se pretender tornar o processo de seleção de conteúdos mais humano, é algo bastante interessante, pois libertaria as pessoas do trabalho de terem que filtrar a informação que lhes chega, também eliminando o problema da sobrecarga de informação irrelevante.

1.5 Objetivos

O objetivo do trabalho é um algoritmo de classificação e seleção de conteúdo (anúncios), tendo em conta os objetivos dos anunciantes (público-alvo, horários de emissão), informação de contexto (programas, localização), e preferências de utilizador (programas preferidos).

Outro objetivo deste trabalho passa pelo desenvolvimento de um *webservice*, com uma interface simples e fácil de usar, acessível por outros serviços, capaz de consultar o estados dos dados (programas e anúncios), fazer a sua gestão e ainda escolher anúncios com base nos critérios já enunciados.

O desenvolvimento do algoritmo e respetiva integração implica:

- Investigação do estado da arte ao nível de algoritmos já usados (classificação, extração de regras e métricas para cálculo de semelhança);
- Desenvolvimento do algoritmo de classificação e estabelecimento de métricas a serem usadas;
- Desenvolvimento do algoritmo para extração de regras de associação entre programas e anúncios;
- Integração do algoritmo num *webservice*, capaz de satisfazer pedidos do exterior;
- Validação de resultados.

1.6 Estrutura da Dissertação

Este relatório, para além da introdução, apresenta mais 5 capítulos.

No capítulo 2, é descrito o estado da arte das matérias de algoritmos de classificação, métricas usadas para cálculo de semelhanças e ainda algoritmos de extração para a construção de regras de associação. É também feita uma breve análise a serviços de distribuição de publicidade contextualizada.

No capítulo 3, é apresentado o conceito de matching e analisado o problema de forma detalhada, programas e anúncios, assim como o conjunto de critérios que constituem uma parte fulcral

para a seleção do anúncio. Foi também desenvolvida uma metodologia que descreve a abordagem realizada, a divisão por fases, pré-processamento e *matching* (*offline* e *online*), e fluxo de informação.

No capítulo 4, é apresentada a arquitetura do sistema, física e lógica, assim como a estrutura da base de dados relacional usada. É ainda feita uma análise ao esforço de desenvolvimento que existiu ao longo do período de desenvolvimento.

No capítulo 5, são apresentados resultados obtidos por forma a analisar o desempenho e a eficiência do processo de *matching* e seleção de anúncios. São ainda comparados alguns serviços descritos na revisão bibliográfica, de forma qualitativa, com o serviço desenvolvido.

Finalmente, no último capítulo, o capítulo 6, são apresentadas as conclusões tiradas ao longo desta fase, e ainda algumas perspetivas de trabalho futuro.

Capítulo 2

Revisão Bibliográfica

Entre os objetivos desta dissertação, encontra-se o desenvolvimento de um algoritmo que seja capaz de escolher um determinado anúncio, a partir de informações de contexto que caracterizem o ambiente em que um utilizador se encontra. Como qualquer algoritmo de classificação, é também necessário a existência de boas métricas de forma a obter uma boa classificação.

As próprias tendências dos anunciantes são importantes e a inferência de associações entre programas e anúncios torna-se um critério importante ao classificar anúncios.

Neste capítulo são apresentadas revisões do estado da arte ao nível de algoritmos para extração de regras de associação e de classificação, assim como medidas de semelhança. É ainda feita uma análise a serviços de distribuição de publicidade contextualizada

2.1 Classificação de Conteúdos

A classificação de objetos é algo que faz parte da condição do ser humano, tentando sempre comparar algo novo com conhecimentos do mundo que já possuía. Este processo envolve que a pessoa saiba como avaliar os objetos, recorrendo a técnicas de medição de semelhança. Ao longo dos tempos já tem havido muitas abordagens ao problema da classificação de conteúdos e os métodos para o fazer de forma eficiente.

Nesta secção são revistos alguns desses métodos, usados para classificar objetos assim como as medidas de semelhança aplicadas.

2.1.1 Algoritmos de Classificação

O objetivo dos algoritmos de classificação é prever o valor de uma determinada variável (dado o alvo ou objetivo). Se a variável-alvo for categórica, é um problema de classificação e, se for numérica, é um problema de regressão. Para cada registo no conjunto de dados, é determinado o valor da classe atributo. Assim, é construído o modelo baseado num conjunto de treino, sendo usado para prever novos dados.

Estes algoritmos são construídos com um processo de duas fases [kn:12g]. A primeira fase é a construção do modelo, em que, para cada amostra (ou registo, ou objeto, ou instância, ou exemplo), se assume que tem um valor para o atributo-alvo. O conjunto de amostras usado para a construção do modelo designa-se de conjunto de treino. O modelo é representado como regras de classificação, árvores de decisão, ou fórmulas matemáticas. A segunda fase do processo é a utilização do modelo para prever objetos futuros ou desconhecidos. É estimada a precisão do modelo comparando o resultado obtido com os resultados esperados, conhecidos previamente. Para além dos modelos de treino e de teste, pode também ser utilizado um terceiro conjunto, o conjunto de poda, que constitui 30% do conjunto de treino, de forma a eliminar os subconjuntos menos frequentes.

Ao longo da classificação, poderá surgir ainda um problema de sobre adaptação (*overfitting*). Este ocorre quando o modelo adaptou-se demasiado ao conjunto de treino, tornando assim a sua capacidade de prever bastante fraca, podendo surgir erros aleatórios ou ruído em vez de relações. Por outras palavras, o modelo memorizou o conjunto de treino em vez de aprender a generalizar a partir das tendências [kn:12g].

2.1.1.1 *OneR*

O *OneR* trata de aprender uma árvore de decisão de um nível apenas, ou seja, as regras que classificam um objeto são baseadas num único atributo. A versão básica deste algoritmo contém um ramo para cada valor, em que cada ramo está atribuído a uma classe frequente. O atributo escolhido é aquele com a menor taxa de erro (proporção de instâncias que não pertencem à classe maioritária do seu ramo correspondente).

Ao lidar com atributos numéricos, deve-se dividir a variação de cada atributo em intervalos e distribuir as instâncias consoante os valores. Ao separar as classes desta maneira, o erro total é diminuído [kn:12g].

2.1.1.2 Classificadores *Bayesianos*

Este algoritmo baseia-se no teorema de Bayes em que, dado um conjunto de treino D, a probabilidade posterior de uma hipótese h, P(h|D) é dada pela fórmula

$$P(h|D) = P(D|h).P(h)/P(D) \tag{2.1}$$

A classificação é feita com base na suposição que todos os atributos são igualmente importantes e estatisticamente independentes (o valor de um, nada diz sobre outro). O objetivo deste algoritmo é então identificar as hipóteses com o máximo valor de probabilidade posterior. Contudo, uma das dificuldades que se impõe neste algoritmo é o facto de exigir ter conhecimento de demasiadas probabilidades, implicando um custo computacional muito elevado. O problema de classificação pode ser formalizado usando probabilidades a-posteriori. Consideremos P(C|X) = probabilidade que o registo X = <x1,x2,. . . .,xk> seja da classe C.

$$P(C|X) = P(X|C).P(C)/P(X) \tag{2.2}$$

Contudo, o cálculo de P(X|C) = P(x1,x2,...,xk|C) é inviável, levando a recorrer a outras formas de cálculo. Uma dessas formas é o algoritmo de classificação Bayesiana ingénuo. Este tipo de classificação Bayesiana rege-se pelo princípio de que todos os atributos são independentes, reduzindo assim substancialmente o custo computacional. Se o atributo i for categórico, P(Xi|C) é estimado como a frequência relativa de amostras contendo o valor Xi como atributo da classe C. Caso i seja contínuo, P(Xi|C) é estimado através da função de densidade Gaussiana.

Este tipo de algoritmo funciona bastante bem (mesmo que a suposição de independência seja violada). A classificação não necessita estimativas probabilísticas precisas, desde que a probabilidade máxima esteja atribuída à classe correta. Contudo, adicionar demasiados atributos redundantes irá causar problemas (atributos idênticos), e também muitos atributos numéricos não estão distribuídos normalmente [kn:12g].

2.1.1.3 *K-Nearest Neighbor*

O K-NN trata-se de um algoritmo que aprende baseado em instâncias, cuja aprendizagem consistem em armazenar todas as instâncias de treino, e a classificação resume-se a atribuir a função alvo a uma nova instância. É por isso chamado de aprendizagem preguiçosa (*lazy learning*).

Os classificadores de vizinho mais próximo (*nearest neighbor*) são baseados na aprendizagem por analogia. Todas as instâncias correspondem a pontos num espaço de n dimensões. O vizinho mais próximo é definido em termos da distância euclidiana.

Para valores discretos, o K-NN devolve o valor mais comum entre os K exemplos de treino mais próximos de xq. Caso o valor de k seja igual a 1, o classificador será instável e, ao mesmo tempo, se for muito grande significa que os pontos não estão muito próximos.

Pode ser atribuído um peso às distâncias dos vizinhos, ou seja, a pesar a contribuição de cada um de acordo com a distância deles a uma interrogação, dando valor mais alto aos vizinhos mais próximos [kn:12g].

2.1.1.4 Árvores de Decisão

Aprendizagem por árvores de decisão é usada como modelo preditivo capaz de fazer conclusões, a partir de observações. Cada nó corresponde a um teste sobre um atributo, cada ramo a um valor e cada folha corresponde a uma classe. Um caminho é um conjunto de valores de atributos.

A grande vantagem de usar árvores de decisão prende-se pelo facto da sua precisão ser comparável, ou até superior a outros métodos, sendo também rápidas de construir e facilmente assimiláveis pelos humanos. A geração destas é constituída por duas fases, a sua construção e pela fase de poda (*pruning*). Na primeira, todos os exemplos de treino estão na raiz e é particionada, recursivamente, baseando-se nos atributos selecionados. Já a segunda fase, identifica e remove os ramos que refletem "ruído". Este tipo de árvores lida bem com conjuntos de dados elevados. O

problema destas árvores é que, quanto mais ramos, isto é, mais complexa, mais frequente será o fenómeno de *overfitting* [kn:12g].

2.1.2 Medidas de Semelhança

Os algoritmos que acabaram de ser analisados permitem classificar dados e agrupá-los. Contudo, necessitam de ter métricas que permitam medir a semelhança entre objetos, isto é, são precisas maneiras de saber o quão parecidos ou diferentes são em comparação um com o outro. No final do processo de classificação, dentro de cada grupo os objetos são semelhantes entre si e diferentes entre grupos, daí referir-se a essa relação através de uma medida de proximidade.

A medida de semelhança entre dois objetos será zero, caso os objetos não sejam parecidos e, quanto maior o valor, mais parecidos são. A medida de dissemelhança funciona da maneira oposta, ou seja, quanto maior o valor, menos parecidos são, enquanto se o valor for zero, os objetos são semelhantes entre si. A medida de semelhança e dissemelhança entre dois objetos i e j pode ser calculadas segundo a fórmula seguinte:

$$sim(i,j) = 1 - d(i,j) \tag{2.3}$$

Esta secção apresenta medidas de semelhança e dissemelhança para objetos com atributos nominais, binários e numéricos, e ainda de comparação de palavras.

2.1.2.1 Atributos Nominais

Um atributo nominal é um atributo que pode tomar dois ou mais estados como, por exemplo, a cor de um objeto que pode ser azul, vermelho, verde, etc. A medida de dissemelhança [HKP06] entre dois objetos i e j pode ser calculada com base na proporção de incompatibilidades:

$$d(i,j) = \frac{p-m}{p} \tag{2.4}$$

Nesta fórmula, m representa o número de correspondências, isto é, o número de atributos em que i e j têm o mesmo estado, e p representa o número total de atributos que descrevem os objetos. Podem ser atribuídos pesos aos atributos de forma a realçar os pontos em comum. A semelhança pode ser também calculada a partir da seguinte maneira:

$$sim(i,j) = 1 - d(i,j) = \frac{m}{p} \tag{2.5}$$

Os atributos nominais podem ser interpretados de uma maneira binária, ou seja, para cada estado de cada atributo o valor que representava esse estado é 1, enquanto os restantes estados do atributo assumiam o valor 0. Por exemplo: O atributo cor de um objeto representada pelo formato

RGB (vermelho, verde, azul), tem como estado azul, ou seja, podia ser interpretado de forma binária para 001.

2.1.2.2 Atributos Binários

Um atributo binário é aquele que apenas pode assumir dois estados: 0, em que o atributo não existe, e 1, em que o atributo está presente. Uma maneira de calcular a dissemelhança entre dois atributos binários envolve a construção de uma matriz de dissemelhança. Considerando que todos os atributos têm o mesmo peso, obtém-se uma tabela 2x2 em que q representa o número de atributos de valor igual a 1 para ambos os objetos, r é o número de atributos de valor 1 para i mas 0 para j, s é o contrário, ou seja, representa o número de atributos de valor 0 para i mas 1 para j e, finalmente, t é o número de atributos de valor 0 para ambos os objetos [HKP06].

		Objeto j		
Objeto i	-	1	0	sum
	1	q	r	q + r
	0	s	t	s + t
	sum	q + s	r + t	p

Tabela 2.1: Extração de regras

Nesta situação, a medida de dissemelhança entre dois objetos i e j pode ser obtida pela seguinte fórmula.

$$d(i,j) = \frac{r+s}{q+r+s+t} \tag{2.6}$$

No caso de atributos assimétricos, isto é, que não têm o mesmo peso, a medida de dissemelhança é calculada ignorando o valor de t, visto o resultados negativos não interessarem para este caso.

$$d(i,j) = \frac{r+s}{q+r+s} \tag{2.7}$$

Coeficiente de semelhança de Jaccard É também possível medir a diferença entre dois atributos binários baseando-se no conceito de semelhança, como é o caso do coeficiente de semelhança de Jaccard [HKP06, Jac01]. Este coeficiente é capaz de medir a semelhança entre dois objetos, tendo em conta tanto os objetos em comum como aqueles em que diferem permitindo assim que tanto os atributos simétricos (mesmo peso) ou assimétricos (pesos diferentes) possam aparecer, segundo a fórmula:

$$sim(i,j) = \frac{q}{q+r+s} = 1 - d(i,j) \tag{2.8}$$

Distância de Hamming A distância de Hamming [Ham50] calcular o número de diferenças entre dois objetos a partir do número de atributos diferentes, logo quanto maior for esse valor menos parecidos serão esses objetos. A mesma técnica é aplicada a conjuntos binários, em que o número de 1's define o valor da dissemelhança desses objetos, através da aplicação de um operador lógico XOR[1].

2.1.2.3 Atributos Numéricos

Atributos numéricos, como o próprio nome indica, são atributos que podem assumir qualquer valor numérico. Nalguns casos, os dados costumam ser normalizados antes de aplicar qualquer cálculo de distância. A normalização envolve transformar o conjunto de valores para um intervalo mais reduzido, como por exemplo [0,1], de forma a simplificar os cálculos que se irão seguir. Há duas maneiras de o fazer, em que a primeira é necessário saber qual o valor máximo do intervalo em que os valores podem variar, calculando depois a proporção de cada. Já a segunda maneira, admite o valor máximo do conjunto como limite superior do intervalo, substituindo-o por 1, e calcula os restantes relativamente a esse.

$$[5,10,6,8] \Rightarrow [0.5,1,0.6,0.8] \tag{2.9}$$

Distância Euclidiana Esta medida de distância [HKP06] é a mais conhecida que consiste em medir a distância mais curta entre dois pontos i e j. Considerando o conjunto $i = (x_{i1}, x_{i2}, ..., x_{ip})$ e $j = (x_{j1}, x_{j2}, ..., x_{jp})$ e p o número de atributos, a distância entre os objetos i e j é:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2} \tag{2.10}$$

Distância de Manhattan A distância de Manhattan [HKP06], também conhecida como distância "quarteirão" (city block) devido ao facto do cálculo da distância entre dois pontos ser calculada em blocos entre cada dois pontos de uma cidade (ex: 1 bloco para cima e 3 blocos para a direita), é outra medida de cálculo da distância entre objetos com atributos numéricos, podendo ser obtida pela fórmula.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}| \tag{2.11}$$

Distância de Minkowski A distância de Minkowski [HKP06] trata-se de uma generalização das distâncias Euclidianas e de Manhattan.

[1] O operador lógico XOR, devolve o valor 0 quando todos os valores de entrada são iguais (tudo 0's ou tudo 1's), e 1 quando apenas um dos valores de entrada é diferente dos outros

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + ... + |x_{ip} - x_{jp}|^h} \tag{2.12}$$

O valor de h é um número real e quando h=1 obtém-se a distância de Manhattan e h=2 a distância Euclidiana.

Distância de Chebyshev A distância de Chebyshev [HKP06] trata-se de uma generalização da distância de Minkowski para $h \to \infty$, daí também ser designada por distância suprema (supremum distance). A maneira de calcular é descobrindo o atributo f que dá a maior diferença de valores entre os dois objetos.

$$d(i,j) = \lim_{h\to\infty} (\sum_{f=1}^{p} |x_{ip} - x_{jp}|^h)^{\frac{1}{h}} \tag{2.13}$$

2.2 Extração de Regras de Associação

A extração de regras de associação consiste em encontrar padrões frequentes, associações ou correlações entre conjuntos de itens em bases de dados que contêm transações (por exemplo, perceber os hábitos de compra de um cliente). Uma transação é uma lista de itens associados a uma entidade. As regras de associação costumam ter um antecedente e um consequente, ou seja, a presença de um ou mais itens (antecedentes) implica a presença de outro ou outros itens (consequentes):

$$Antecedente \Rightarrow Consequente[s,c] \tag{2.14}$$

Estas regras são expressas também com um nível de suporte (s) e confiança (c). O suporte denota a frequência da regra dentro das transações. Um valor alto significa que a regra envolve uma grande parte da base de dados.

$$s(A \Rightarrow B[s,c]) = p(A \cup B) \tag{2.15}$$

A confiança define a percentagem de transações que contêm A, mas também contêm B. É uma estimativa da probabilidade condicionada.

$$c(A \Rightarrow B[s,c]) = p(B|A) = s(A,B)/s(A) \tag{2.16}$$

Hoje em dia, são conhecidos bastantes algoritmos capazes de criar estas regras de associação através da análise à base de dados, podendo usar estas medidas, suporte e confiança, para avaliar regras e identificar quais são as interessantes.

2.2.1 *Apriori*

O algoritmo *Apriori* [kn:12g, AS94] é um dos algoritmos usados para a extração de regras de associação. A sua abordagem consiste em gerar candidatos (conjuntos) e encontrar (testar) todos os conjuntos de itens que têm elevado suporte (conjuntos frequentes) e gerar regras de associações a partir destes. Este algoritmo rege-se pelo princípio de que qualquer subconjunto de um conjunto frequente também o é, e que o contrário também se verifica, isto é, nenhum superconjunto de um conjunto que não seja frequente deve ser gerado ou testado.

O passo principal neste algoritmo trata-se de encontrar o conjunto frequente de itens, ou seja, aqueles que têm (pelo menos) suporte mínimo. Quanto à geração de candidatos, o algoritmo *Apriori* faz uma abordagem de baixo para cima (*bottom-up*), gerando conjuntos de itens candidatos com comprimento de (k+1), em que k é o comprimento dos conjuntos frequentes de itens, e testando os candidatos com a base de dados para determinar os que serão de facto frequentes.

O segundo passo da geração de candidatos envolve uma operação de poda (*pruning*). Nesta fase, qualquer conjunto de itens (k-1) que não seja frequente, não pode ser um subconjunto de um conjunto frequente k. Esta operação envolve contar o suporte para cada conjunto e é uma operação exigente, em termos computacionais, então pode-se recorrer a técnicas de eliminar candidatos mais leves como combinar candidatos e ver se os seus subconjuntos existem.

2.2.2 *TreeProjection*

O algoritmo *TreeProjection* [AAP00] parte do princípio que existem uma ordem lexicográfica entre os itens de uma base de dados e consiste na construção de uma representação estrutural dos conjuntos de itens frequentes que respeitam esta ordem, em forma de árvore. Cada ramo corresponde a uma extensão e cada nível corresponde aos tamanhos dos diferentes conjuntos de itens. Cada nó diz-se gerado, a primeira vez que é descoberto, pela extensão do nó pai. Ao contrário do método *Apriori* [AS94], a construção da árvore é feita incrementalmente, algo que reduz substancialmente os custos computacionais visto não haver uma fase de geração e teste de candidatos.

Cada nó contém as projeções das suas extensões, daí o nome *TreeProjection*, pois desta maneira ajuda a reduzir o tempo de CPU quando for a contar o suporte, para cada conjunto de itens. Estas projeções são armazenadas na forma de uma matriz triangular.

As árvores são construídas recorrendo a métodos de pesquisa em largura, pesquisa em profundidade ou combinação dos dois. No modo de pesquisa em largura, todos os nós de um nível k são gerados antes dos nós do nível (k + 1), e é necessário iniciar o processo de projeção de transações em cada iteração, começando do nó-raiz. Já modo de pesquisa em profundidade, os nós são criados até atingir a sua profundidade máxima para cada nó pai. Ao contrário do modo de pesquisa

em largura, não é necessário iniciar o processo de projeção de transações em cada iteração, pois apenas é necessário percorrer a base de dados apenas uma vez, o que também pode ser considerado um problema caso a base de dados seja bastante grande, podendo originar problemas de memória, pois a qualquer altura, são mantidas projeções para todos os nós (e seus irmãos), no caminho que vai desde o nó raiz até ao nó que está atualmente a ser expandido.

2.2.3 *FP-Growth*

O *FP-Growth* [HPY00], *Frequent Pattern Growth*, à semelhança do *TreeProjection*, consiste na construção de uma estrutura de dados em forma de árvore de forma iterativa, que armazena informação quantitativa de padrões frequentes. Uma das características desta árvore, de forma a garantir que a sua estrutura seja compacta e informativa, é que em cada nó, é armazenado um item de comprimento-1, sendo cada conjunto de itens frequentes representado por uma sequência (caminho) de nós.

As principais operações do processo de extração são contar acumulações de itens e ajustar as contagens, sempre que um novo conjunto de itens frequente é adicionado. Estas operações são muito menos custosas que geração e teste de candidatos como o *Apriori*, também sendo mais eficiente que armazenar projeções para cada nó, como o *TreeProjection* [HPY00].

O procedimento do algoritmo consiste em contar os itens existentes na base de dados, a sua frequência e excluir (*pruning*) os que não têm suporte mínimo. Em seguida, ordena os itens frequentes encontrados, de cada transação, e constrói a respetiva árvore.

2.2.4 H-Mine

O H-Mine [PHL+01] é um método usado para extração de informação que aproveita diferentes abordagens de outros algoritmos, para construção de uma estrutura de dados designada H-Struct.

Tal como o FP-Growth e TreeProjection, existe um pré-processamento da base de dados de transações, que consiste em contar os itens frequentes pois, seguindo a propriedade do Apriori, apenas itens frequentes serão relevantes. Após este processamento, a base de dados de transações encontra-se apenas com os itens frequentes (com frequência superior ao suporte mínimo), e ordenados [PHL+01].

Cada ocorrência de um item é armazenado numa entrada com dois campos, o seu identificador e uma ligação para outro item e estas são usadas para construir uma tabela em que cada entrada tem três campos: o identificador, o valor de suporte desse item e a ligação. As projeções de itens frequentes são carregadas, e as transações que começam com os mesmos itens são interligadas entre si, no que irá constituir a H-Struct. Todo o resto do processo é executado sobre a H-Struct [PHL+01] já não havendo necessidade de analisar a base de dados de transações.

A extração é feita percorrendo os elementos da tabela, em que cada fila ligada aos itens interligados é tratada como um sub-H-Struct e são aplicadas as técnicas recursivamente, isto é, são

procurados os itens locais frequentes, volta-se a particionar o subconjunto de itens frequentes resultante e realiza-se extração recursiva.

2.3 Serviços de Publicidade Contextualizada

A publicidade é um meio bastante influente no que toca a pessoas, podendo fazer a diferença entre comprar um produto ou até chegar mesmo a não querer ter nada haver com ele. Os anunciantes tentam sempre explorar a melhor maneira para utilizar esta ferramenta, arranjando sempre formas de reinventar os seus anúncios e até a maneira de eles chegarem às pessoas.

Os serviços de publicidade contextualizada ganham aqui uma nova dimensão, pois permitem aos anunciantes selecionar uma série de critérios que definem o tipo de público-alvo que pretendem para os seus anúncios.

Nesta secção são apresentados alguns serviços que disponibilizam publicidade baseada em informação de contexto.

2.3.1 *Advertising.com*

A empresa *Advertising.com* [kn:12a] é uma companhia de publicidade online fundada em 1998 nos Estados Unidos da América, que se dedica a distribuir personalizada utilizando para isso diferentes critérios, definidos pelos anunciantes, para a escolha do público-alvo a que os anúncios deverão ser exibidos.

Os critérios que estão disponíveis são:

- **Comportamento** - este critério define o público-alvo com base no seu comportamento online, podendo ser personalizado a diferentes segmentos;
- **Residencial** - permite direcionar anúncios para pessoas que tenham hábitos de compra para certos produtos (informação obtida através de inquérito) e estilos de vida pretendidos;
- **Audiências modeladas** - o anúncio é exibido a pessoas que apresentem semelhanças com públicos-alvo pretendidos;
- **Redirecionamento** - a partir de dados históricos que demonstram que certas audiências estão interessadas nos seus produtos, a publicidade pode ser redirecionada para os mesmos;
- **Demográfica** - informações como o sexo, idade, rendimentos, crianças, etc. influenciam a exibição do anúncio;
- **Geográfica e temporal** - permite definir locais e horas do dia em que os anúncios devem ser passados;
- **Tecnográfica** - utiliza informações da tecnologia que o computador dos utilizadores utiliza para acesso à rede como critério para seleção de público-alvo.

2.3.2 *Clicksor.com*

A *Clicksor.com* [kn:12b] é uma rede de distribuição de publicidade de contexto, que oferece soluções no que toca a distribuir conteúdos publicitários, a públicos-alvos definidos, em tempo real.

Tal como outros serviços permite ao anunciante definir um conjunto de critérios que definem o tipo de público-alvo para os seus anúncios. Estes critérios são:

- **Geográfica** - controla a localização (país) onde o anúncio deve ser exibido;
- **Canal** - permite definir o canal onde o anúncio pode ser exibido;
- **Tempo** - controla as horas do dia em que o anúncio pode ser passado;
- **Palavras-chave** - permite selecionar um conjunto de palavras-chave que ficarão associadas ao anúncio;
- **Redirecionamento** - a partir de dados históricos que demonstram que certas audiências estão interessadas nos seus produtos, a publicidade pode ser redirecionada para os mesmos;
- **Sistema Operativo** - o sistema operativo (Windows, Mac OS, Linux) define o público-alvo a receber um anúncio.
- **Dispositivos** - o tipo de dispositivos (portátil, telemóvel, *smartphone*, *tablet*) define o público-alvo daquele anúncio;
- **Serviço de Internet** - o público-alvo é escolhido pelo serviço de Internet que usa;
- **Linguagem** - permite definir o público-alvo consoante a sua língua, de forma a atingir um alcance global.

2.4 Sumário

Ao longo deste capítulo foram introduzidos os conceitos de classificação de conteúdo, as suas vantagens e usos, assim como uma breve análise a algoritmos que já tratam deste tipo de problema. Os algoritmos analisados dependem fortemente do seu contexto, o *OneR* será bom para classificação baseada em apenas um atributo, enquanto que, se houverem mais atributos, deve-se ter em conta outros como os classificadores *bayesianos*, quando atributos estão dependentes de outros, árvores de decisão, quando se pretende atingir velocidade de construção da árvore, e K-NN, quando atributos diferentes têm pesos diferentes na classificação.

O tipo de dados a serem usados para estabelecer as ligações influencia bastante a maneira de classificar conteúdos, daí também haver necessidade de definir métricas que possam ser usadas para calcular a semelhança ou dissemelhança entre objetos, de forma a poder agrupá-los.

A extração de regras revelou-se também bastante importante como critério para identificação de padrões e ate mesmo, podendo ser usado como critérios de classificação. De entre os algoritmos

revistos, o *H-Mine* mostra-se mais adequado que os restantes, sendo mais rápido que o *Apriori*, por este perder demasiado temppo a gerar candidatos, e mais eficiente que o *TreeProjection* e *FP-Growth*, quando a base de dados é grande e dispersa.

Ainda neste capítulo, também é apresentado o conceito de serviço de publicidade contextualizada, que aproveita informações de contexto definidas por anunciantes, para fazer corresponder àquelas que definem o ambiente em que os utilizadores se encontram a determinado momento. Os serviços existentes, como *Advertising.com* e *Clicksor.com*, oferecem bastantes soluções aos anunciantes, algumas únicas e outras comuns, de forma a que estes possam definir o público-alvo dos seus anúncios.

Capítulo 3

Modelação do Problema

Hoje em dia, a televisão, mais que um instrumento de lazer, serve também para comunicar e informar. Os anunciantes exploram este objeto, algumas vezes de forma excessiva, fazendo com que o utilizador chegue mesmo a afastar-se da televisão durante os intervalos de programas, não visualizando os anúncios e fazendo com que os anunciantes não cumpram o seu principal objetivo: publicitar novos produtos.

Surgiu portanto, no âmbito desta tese, a necessidade de desenvolver um algoritmo capaz emparelhar programas e anúncios – *matching* –, tendo em conta as preferências dos anunciantes, permitindo que sejam passados aquando das visualizações do próprios programas e utilizando outras informações de contexto, nomeadamente preferências de programas dos utilizadores, a sua localização e ainda dia e hora da semana, por forma a garantir que os anúncios sejam visualizados pelo público certo.

3.1 Programas

Os programas são os conteúdos que irão permitir que os anúncios sejam visualizados, isto é, são estes conteúdos televisivos que dão começo ao processo de seleção dos anúncios, pois o objetivo é que o anúncio seja passado antes, durante (nos intervalos) ou no final de cada programa. Os programas são classificados com base no seu formato e géneros associados.

Cada programa tem apenas um formato associado, mas podendo ter vários géneros. A única restrição existente, é que os géneros estejam em conformidade com os formatos a que pertencem.

3.1.1 EPG

O *Electronic Programming Guide*, também conhecido como EPG, é uma interface gráfica da televisão digital, capaz de disponibilizar informação sobre os programas que irão passar em cada canal, tendo ainda várias outras funcionalidades. No âmbito deste projeto, é usado o EPG do SAPO para obter a lista dos programas a serem classificados e armazenados.

MOVIES	SERIES	TALK-SHOW	HUMOR	REALITY SHOW	TALENT SHOW	SOAP OPERA	SPORTS	NEWS	MUSIC
ACTION	ACTION	LATE NIGHT	HIDDEN CAMERA		ADVENTURE	COMEDY	BASKETBALL	CULTURE	ERUDITE
ADVENTURE	ADVENTURE	LIFESTYLE	STAND-UP		COOKING	DRAMA	COMBAT SPORTS	ECONOMICS	GENERAL MUSIC
ANIMATION	ANIMATION				DANCE	JUVENILE	CYCLING	MAGAZINE	
BIOGRAPHY	BABIES				LIFESTYLE		EXTREME	POLITICAL	
COMEDY	BIOGRAPHY				MUSIC		FOOTBALL		
CRIME	COMEDY						GOLF		
DOCUMENTARY	COOKING						HANDBALL		
DRAMA	CRIME						INDOOR SPORTS		
FAMILY	DOCUMENTARY						MAGAZINE		
FANTASY	DRAMA						MOTOR SPORTS		
FILM-NOIR	FAMILY						OUTDOOR SPORTS		
HISTORY	FANTASY						SOCCER		
HORROR	FASHION						TENNIS		
MISTERY	HISTORY						WATER SPORTS		
MUSIC	HORROR						WINTER SPORTS		
MUSICAL	KIDS								
ROMANCE	LIFESTYLE								
SCI-FI	MEDICAL								
SPORTS	MISTERY								
SUPERNATURAL	MUSIC								
THRILLER	MUSICAL								
WAR	PROCEDURAL								
WESTERN	ROMANCE								
	SCI-FI								
	SPORTS								
	SUPERNATURAL								
	THRILLER								
	WAR								
	WESTERN								

Figura 3.1: Taxonomia de programas

O motivo da escolha deve-se ao facto deste serviço estar sincronizado com o guia de programação do MEO [kn:12c], isto é, os programas identificados pelo EPG do SAPO [kn:12f] são os mesmos usados pelo MEO.

3.1.2 *CineFetch*

Um dos fatores cruciais para este projeto baseia-se no pressuposto que os dados, neste caso os programas, estejam devidamente classificados de acordo com a taxonomia definida na figura 3.1. Contudo, apesar do EPG do SAPO fornecer a lista de programas estes não estão classificados, havendo necessidade de os classificar.

É aqui que surge o *CineFetch*, uma aplicação desenvolvida na PT Inovação, capaz de reunir informação de filmes, séries e outros programas de vários locais da Internet numa base de dados semântica, podendo classificar conteúdos de forma automática e dinâmica de acordo com a taxonomia.

```
1 <program name="Nikita">
2   <format>SERIES</format>
3   <genres>
4     <genre>CRIME</genre>
5     <genre>DRAMA</genre>
6     <genre>ROMANCE</genre>
7   </genres>
8 </program>
```

Listing 3.1: Programa Nikita classificado

3.2 Anúncios

Os anúncios, à semelhança dos programas são classificados com base numa taxonomia pré-definida, também da mesma estrutura que os programas, ou seja, composta por um "valor alto" chamada de macro, e por um "valor baixo" designado de valor, à semelhança dos formatos e géneros, respetivamente (Figura 3.2);

MACRO-CATEGORIAS						
DESPORTO	RESTAURAÇÃO	CIÊNCIA	EDUCAÇÃO/ENSINO	TECNOLOGIA	INFORMÁTICA	EMPREENDORISMO
EMPREGO	EVENTOS/ACONTECIMENTOS/ESPECTÁCULOS	CASA/JARDIM/DECORAÇÃO	BRINQUEDOS/CRIANÇA	ANIMAIS	MÚSICA	MOTORES
GESTÃO	SAÚDE E BEM-ESTAR	ALIMENTAÇÃO	ENTRETENIMENTO/LAZER	JOGOS/VÍDEO JOGOS	LIVROS/LITERATURA	FILMES
SERVIÇOS	MODA E ACESSÓRIOS	COLECIONISMO	ARTE/CULTURA			
VALORES						
CONCRETIZAÇÃO	CONHECIMENTO	AVENTURA	AFETIVIDADE	AGITAÇÃO	ATITUDE	AUSTERIDADE
COOPERAÇÃO	DESEJO	DIGNIDADE	EFICIÊNCIA	ELEGÂNCIA	FANTASIA	FELICIDADE
EQUILÍBRIO	CUIDADO	MUDANÇA	ACOLHEDOR	COMPETIÇÃO	CONTROLO	CRIATIVDADE
IDENTIDADE	IMAGINAÇÃO	INTEGRIDADE	DIVERSÃO	LIBERDADE	MOTIVAÇÃO	ORGANIZAÇÃO
QUALIDADE	ROMANTISMO	FRAQUEZA	TRABALHO	JUVENTUDE	DINAMISMO	

Figura 3.2: Taxonomia de anúncios

A única diferença dos programas para os anúncios é que os valores não têm macros associadas, logo podem ser a associados um ao outro sem restrições.

```
<ad id="62">
  <name>ZARA Home</name>
  <macro>CASA/JARDIM/DECORACAO</macro>
  <values>
    <value>CUIDADO</value>
    <value>MUDANCA</value>
    <value>CRIATIVDADE</value>
    <value>IDENTIDADE</value>
    <value>ORGANIZACAO</value>
  </values>
  (...)
</ad>
```

Listing 3.2: Anúncio ZARA Home

Este pequeno conjunto de metadados apenas trata uma pequena parte do conjunto total que define um anúncio. Tal como foi referido, os próprios anunciantes podem escolher os programas e o tipo de público-alvo para os seus anúncios. Quanto ao primeiro caso, é o chamado de *sponsoring* (patrocínio), em que programas especiais (ex: jogos de futebol) são escolhidos para passar o anúncio.

```
<ad id="62">
  (...)
  <sponsoring bool="true">
    <program>Jogo de Futebol: Benfica X Sporting</program>
  </sponsoring>
  (...)
```

```
</ad>
```

Listing 3.3: Componente Sponsoring do anúncio ZARA Home

Já no segundo caso, existem dois tipos de público-alvo com base em: - Categorias – os anunciantes têm acesso à taxonomia dos programas selecionando os formatos dos programas em que pretendem anunciar, podendo chegar mesmo a escolher os programas dentro do conjunto selecionado.

```
<ad id="62">
  (...)
  <categories bool="true">
    <category>
      <format>SPORTS</format>
      <programs>
        <program similar="true">Pelas casas do Benfica</program>
      </programs>
    </category>
    <category>
      <format>TALENT SHOW</format>
      <programs>
        <program similar="true">Espacos & casa + cartaz</program>
      </programs>
    </category>
  </categories>
  (...)
</ad>
```

Listing 3.4: Componente Categorias do anúncio ZARA Home

- Perfil – ao contrário do critério anterior, a este não interessa o conteúdo que o utilizador está a ver mas antes os que ele costuma ver, isto é, os anunciantes escolhem, de igual forma ao das categorias, através dos formatos os programas onde querem anunciar, só que o anúncio só irá passar nos utilizadores que veem com frequência aqueles programas (programas preferidos) sendo assim independente ao que está atualmente a ver.

```
<ad id="62">
  (...)
  <profile bool="true">
    <category>
      <format>MOVIE</format>
      <programs>
        <program similar="true">A mulher da casa</program>
      </programs>
    </category>
  </profile>
```

```
11   (...)
12 </ad>
```

Listing 3.5: Componente Perfil do anúncio ZARA Home

Para além dos critérios já enunciados, existe ainda mais um subconjunto de metadados que diz respeito à campanha em que o anúncio está inserido, composto pelos seguintes campos:

- Data de início – data de início da campanha, em que o anúncio será exibido;
- Data de fim – data de fim da campanha, a partir da qual o anúncio deixará de ser exibido;
- Impressões – número máximo de visualizações que o anúncio deverá atingir;
- Dias da semana – dias da semana em que o anúncio pode ser exibido. Este campo pode assumir três valores: dias úteis, fins de semana ou todos os dias;
- Intervalos de tempo – períodos do dia em que o anúncio poderá ser exibido;
- Localização – define as cidades em que o anúncio poderá ser passado. Existem três valores possíveis para este campo: o código postal do local onde o anunciante está a criar o anúncio, uma lista de cidades ou então Portugal inteiro.

```
1  <ad id="62">
2    (...)
3    <campaign>
4      <startdate>07-06-2012</startdate>
5      <enddate>16-08-2012</enddate>
6      <impressions>30</impressions>
7      <weekdays>Todos os dias</weekdays>
8      <scheduling>
9        <starthour>12:00</starthour>
10       <endhour>13:00</endhour>
11       <starthour>14:00</starthour>
12       <endhour>16:00</endhour>
13     </scheduling>
14     <location>
15       <location>Porto</location>
16       <location>Vila Nova de Gaia</location>
17     </location>
18   </campaign>
19 </ad>
```

Listing 3.6: Componente Perfil do anúncio ZARA Home

Ao todo, estes dados definem o conjunto de metadados que caracteriza cada um dos anúncios a serem usados pelo algoritmo de *matching* desenvolvido.

3.3 Metodologia

Analisada a estrutura dos metadados dos programas e anúncios, é possível agora estabelecer a metodologia a ser desenvolvida. Esta será composta por duas fases principais, a primeira consiste no pré-processamento dos dados (programas e anúncios) seguido do *matching* entre programas e anúncios, tanto *offline* como *online* (tempo real).

3.3.1 Pré-processamento

Antes de realizar o *matching* é necessário haver dados sobre os quais trabalhar, nomeadamente anúncios e programas.

Quanto aos programas, conforme referido anteriormente, estes são atualizados diariamente a partir do EPG. Contudo, nos dias de hoje, é possível para um utilizador gravar os programas de um dia e visualizá-los mais tarde. Tendo isto em consideração e não querendo sobrecarregar o algoritmo na fase inicial, para além da informação básica de um programa (nome, formato e géneros) é também armazenada a data da última vez que ele foi carregado da lista de programas diários do EPG. Esta data vai servir como validade do programa, ao construir a lista de pares programa-anúncio, sendo atualizada sempre que uma instância do programa esteja presente na lista do EPG (ex: House T.1 EP.22 $\rightarrow$ House). A validade máxima de cada programa é de duas semanas, deixando este de constar da lista de programas candidatos a *matching* quando ultrapassar esse prazo.

Quanto aos anúncios, estes são obtidos de duas maneiras diferentes. A primeira trata-se de fazer *pooling* a um serviço que disponibiliza os metadados respetivos de cada anúncio existente, enviando uma data que servirá para o serviço selecionar apenas os anúncios criados a partir desta. A segunda maneira já é por notificação, por parte do serviço de anúncios sempre que um novo anúncio for criado, acompanhada pelos respetivos metadados.

Terminado o pré-processamento, toda a informação necessária está devidamente atualizada e pronta para ser processada na segunda fase, o processo de *matching*.

3.3.2 Matching

O objetivo principal deste algoritmo é conseguir que devolva sempre um anúncio válido para o tipo de conteúdo que o utilizador está a consumir, neste caso, programas de televisão. Como foi referido anteriormente, os anunciantes têm controlo sobre os vários critérios que definem quando um anúncio deve ou não ser visualizado (dia e hora da semana, programas, público-alvo, localização). Contudo estes critérios não são todos do mesmo tipo, na medida em que alguns são estáticos e outros são dinâmicos.

Os critérios estáticos são aqueles que não alteram com o tempo, ou que as alterações não causam perturbações no funcionamento do algoritmo como é o caso da lista de programas, que poderá ou não ser alterada de um dia para o outro (caso haja novos programas) e, porque a validade de um programa tem uma duração de duas semanas.

Os critérios dinâmicos são os restantes critérios que se vão alterando consoante o dia como é o caso dos horários de emissão, ou que não são possíveis prever com antecedência, como é o caso da localização.

Estes dois tipos de critérios levaram à necessidade do algoritmo se dividir em duas partes, designadas por *matching offline* e *matching online*.

3.3.2.1 *Matching Offline*

O *matching offline* consiste essencialmente em emparelhar anúncios com programas apenas recorrendo aos critérios de *sponsoring*, público-alvo (categorias e perfil) e período da campanha do anúncio.

No *sponsoring*, os anúncios estão destinados a programas especiais e são emparelhados diretamente sem qualquer tipo de cálculo de uma pontuação de semelhança. É o critério mais simples e direto de avaliar (Figura 3.3).

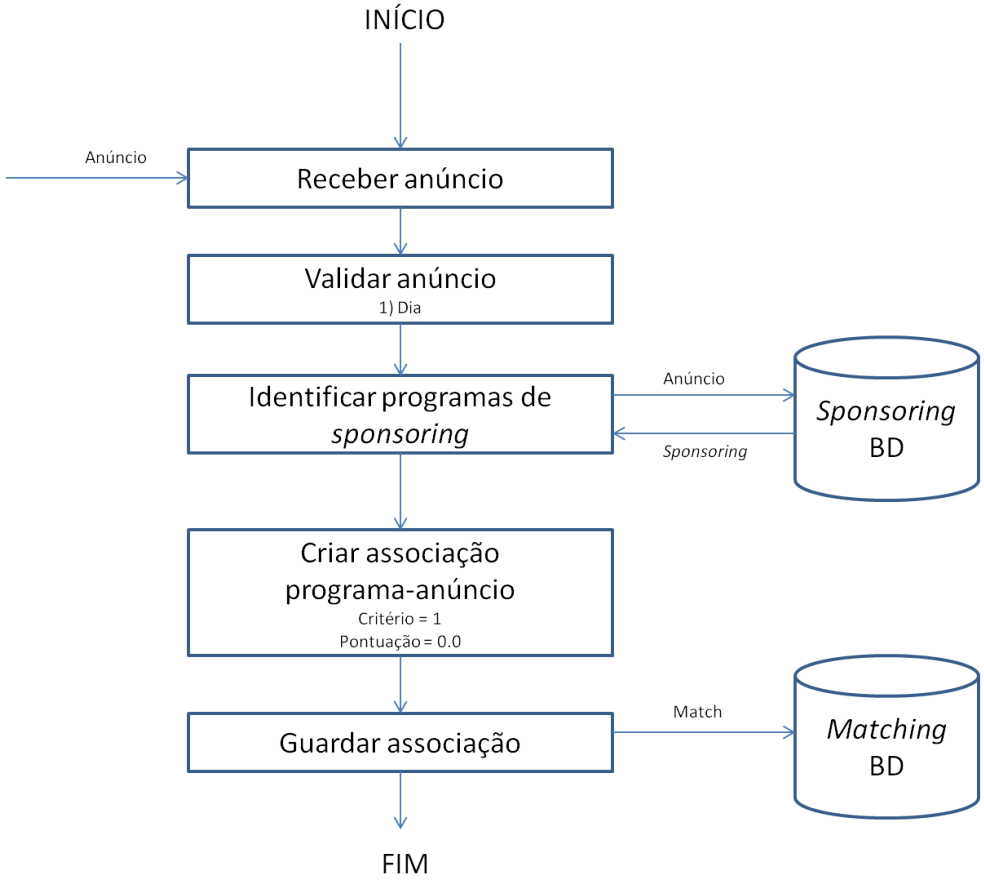

Figura 3.3: *Matching Offline* por *sponsoring*

Quanto às categorias (público-alvo), já se trata de um processo algo mais complexo, relativamente ao *sponsoring*. Os anúncios são alocados aos programas selecionados, sendo ainda calculada uma pontuação para cada par programa-anúncio, construído segundo este critério. Esta

pontuação é calculada com base nos conjuntos formato/géneros do programa e formato/géneros para os quais o anúncio está atribuído neste critério, usando para o efeito a medida de semelhança do coeficiente de Jaccard pois interessa que tanto os pontos em comum como os que não o são influenciem a escolha do anúncio (Figura 3.4).

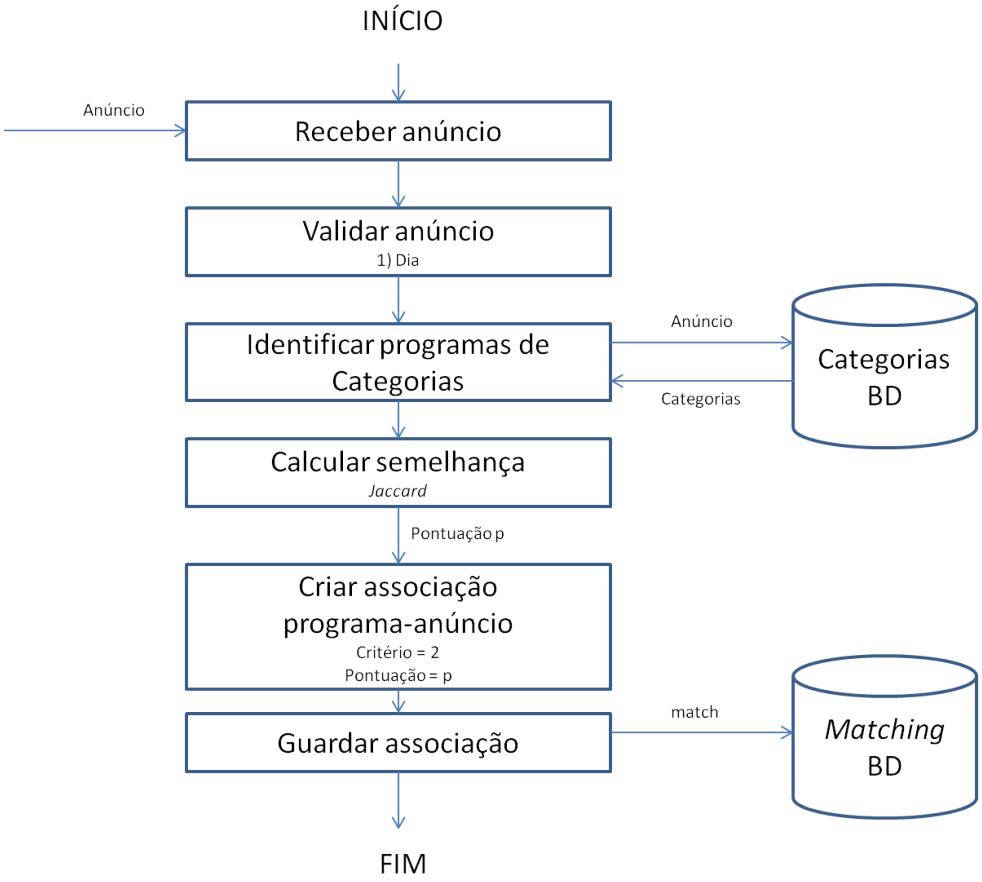

Figura 3.4: *Matching Offline* por categorias

```
1
2  function jaccardCoefficient(Ad ad, Program program}
3
4    // p : Representa o numero de atributos em comum (valor igual a 1 para ambos)
5    // q : Representa os atributos presentes apenas no programa
6    // r : Representa os atributos presentes apenas nos anuncios
7    double p = 0;
8    double q = program.getGenres().size() + 1;
9    double r = ad.getGenres().size() + ad.getFormats().size();
10
11   // Caso haja um formato em comum incrementa p, e decrementa os restantes
12   for(Format f : ad.getFormats())
13     if(program.getFormat() == f) {
```

```
14        p++;
15        q--;
16        r--;
17      }
18
19    // Caso haja um genero em comum incrementa p, e decrementa os restantes
20    for(Genre g : ad.getGenres())
21      if(program.getGenre() == g) {
22        p++;
23        q--;
24        r--;
25      }
26
27    return (p/(p+q+r));
```

Listing 3.7: Coeficiente de Jaccard

O critério do perfil funciona de maneira semelhante ao das categorias, na medida que os anúncios são alocados aos programas selecionados, assim como o método para calcular a pontuação para cada par programa-anúncio. Contudo a forma como o algoritmo irá analisar os anúncios e selecionar já é completamente diferente, como iremos ver mais à frente (Figura 3.5).

Apesar de já vários critérios terem sido definidos para os anúncios, achou-se necessário que fosse criado mais um critério independente do anunciante e que fosse aplicado a todos os anúncios, designado por critério das regras. O único pré-requisito deste quarto critério é que já existam anúncios atribuídos a programas, pois baseia-se na aprendizagem das tendências dos anunciantes, ou seja, com base nas relações formato/géneros e macro/valores já existentes.

O critério das regras é composto por dois passos fundamentais: construção das regras e o respetivo *matching*. O primeiro passo, envolve percorrer todos os pares programa-anúncio já existentes, de forma a extrair os conjuntos formato/géneros e macro /valores mais frequentes. A extração destas associações e a sua construção das regras é feita de forma semelhante ao algoritmo *H-Mine*, em que o antecedente e consequente de cada regra corresponderá aos pares formato/géneros e macro /valores, respetivamente, e o suporte corresponderá à proporção entre a frequência de cada regra face ao total de regras extraídas (Figura 3.6).

O segundo passo é já a construção dos pares programa-anúncio. Para cada combinação possível, verifica-se o conjunto de regras que existem e, caso as haja, a pontuação daquele par será a soma do valor de suporte de cada regra presente (Figura 3.7).

3.3.2.2 *Matching Online*

O *matching online* já lida com os restantes critérios (dinâmicos) e refere-se ao estado do em que o serviço está pronto para receber pedidos de anúncios, necessitando apenas do identificador do programa que está a ser visualizado, fornecido pelo EPG, e o identificador da *MEO Box*, para que o algoritmo consiga processar a informação e encontrar um anúncio adequado. O primeiro passo do algoritmo consiste em obter a lista de anúncios atribuídos na fase anterior (*matching*

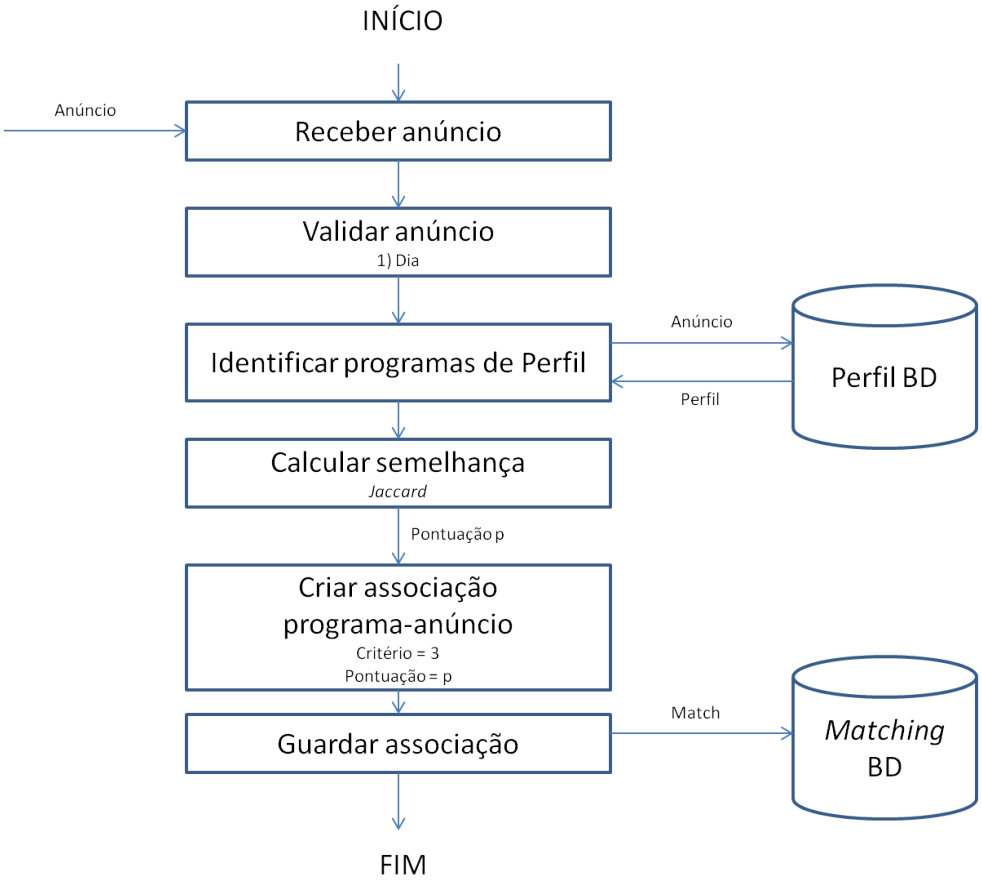

Figura 3.5: *Matching Offline* por perfil

offline) de acordo com a ordem dos critérios estabelecida, sendo o primeiro o critério do *sponsoring*. Cada anúncio obtido será devidamente validado tendo em conta os critérios dinâmicos (dia da semana, hora do dia, número atual de impressões). A localização é também um critério dinâmico, sendo obtida a partir do identificador da *MEO Box* que terá uma posição associada. No final do processo de validação, caso haja mais do que um anúncio válido, recorre-se a critérios de desempate. Neste caso, visto não haver uma pontuação associada entre cada par programa-anúncio, segundo o critério *sponsoring*, usa-se o número de impressões de cada, sendo o que tem mais o escolhido. Isto deve-se pelo facto das impressões serem registadas em contagem decrescente, ou seja, gasta-se o anúncio que ainda tem mais impressões para gastar (o anúncio deixa de ser válido quando chega às zero impressões). Caso ainda exista um empate, será escolhido um anúncio aleatoriamente (Figura 3.8).

Pode ainda acontecer não haver qualquer anúncio válido para aquele programa, tendo o algoritmo que recorrer ao segundo critério, o das categorias. Processando-se de maneira semelhante ao de *sponsoring*, é feita uma validação tendo em conta as variáveis dinâmicas. Contudo, em caso de empate é possível recorrer-se à pontuação calculada anteriormente como critério de desempate, e caso ainda se verifique um empate passa-se ao número de impressões e finalmente o de

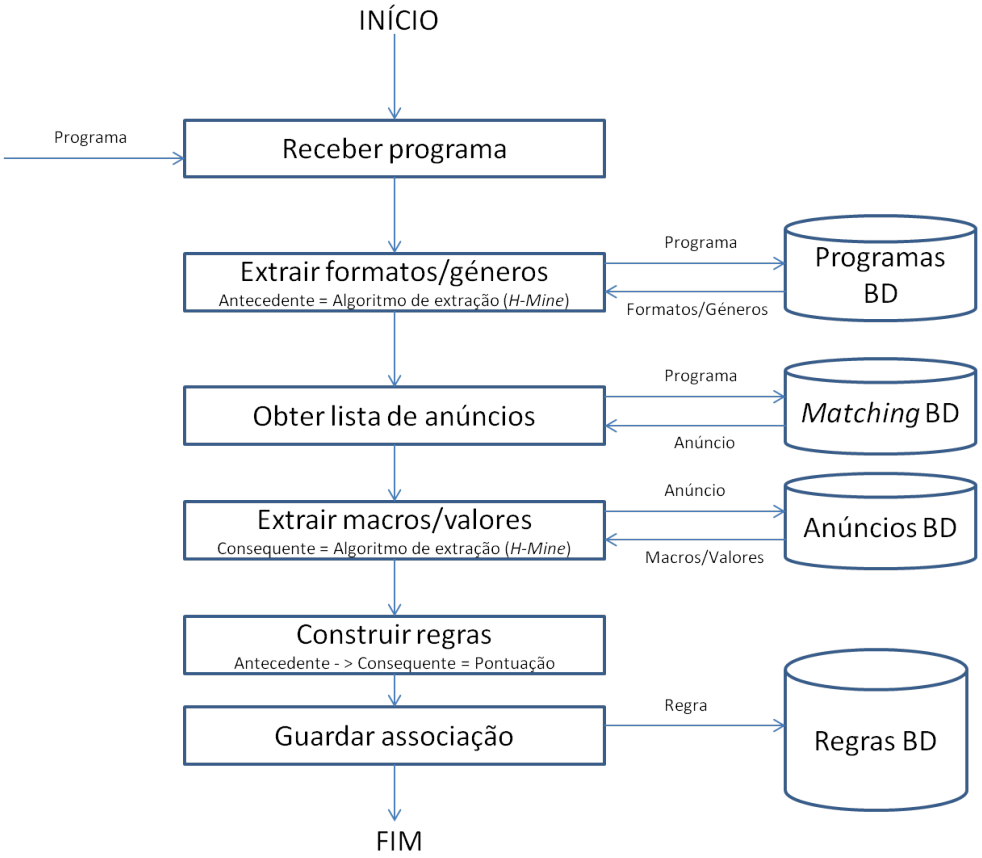

Figura 3.6: Extração de regras

aleatoriedade (Figura 3.9).

Mais uma vez, poderá no final não haver qualquer anúncio válido havendo necessidade de recorrer ao terceiro critério, o de perfil. Como já foi referido anteriormente, para este critério não interessa o conteúdo que está a ser visualizado neste momento, mas sim aqueles que o utilizador mais gosta de ver, mais concretamente os cinco programas preferidos, que podem ser obtidos a partir do identificador da *MEO Box*. Cada programa devolvido desta maneira vem acompanhado de uma pontuação que reflete a sua posição no top. Contudo estes valores não se encontram à mesma escala que as pontuações entre os pares programa-anúncio e portanto é necessário fazer normalização dos valores, em que o maior valor passa a assumir o valor de 1 e os restantes são devidamente normalizados. Seguidamente, para cada programa do top é retirada a lista de anúncios segundo o critério do perfil e recalculada a pontuação de cada par através do produto entre a pontuação normalizada do programa e a pontuação do par. Os anúncios passam então à fase de validação, e em caso de empate, são aplicados os critérios de desempate: pontuação, impressões e aleatório (Figura 3.10).

Apesar destes três critérios tornarem a escolha de um anúncio bastante exaustiva, as variáveis de ambiente podem ainda tornar bastante difícil que um anúncio válido seja escolhido. É aqui que surge o quarto critério, o das regras, construído com base nos primeiros três critérios, serve

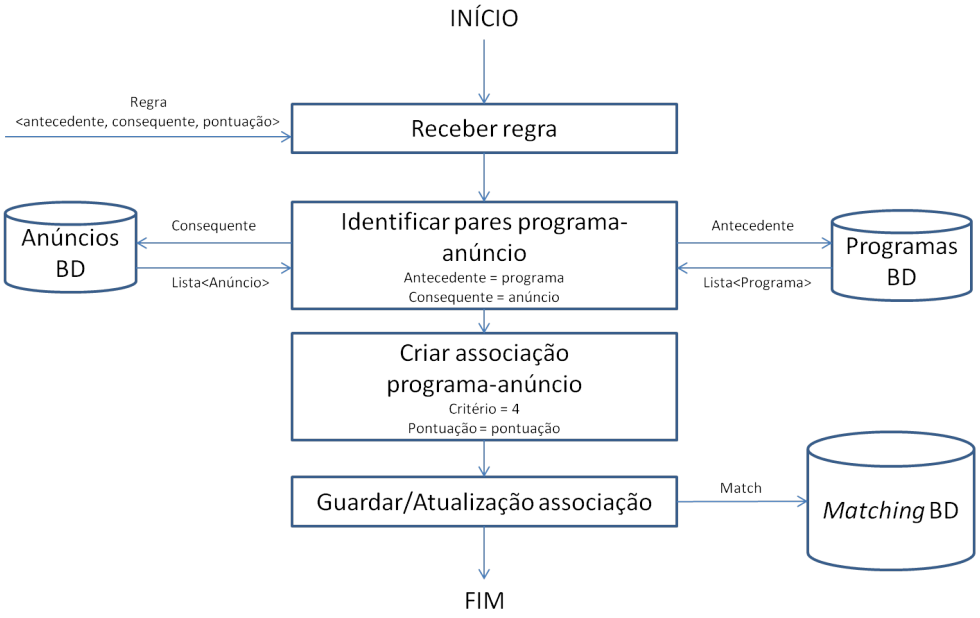

Figura 3.7: *Matching Offline* por regras

para reforçar o processo de escolha. O número de anúncios a processar nesta fase é muito maior, pois todos os pares programa-anúncio existentes, podem chegar a ter apenas uma única regra em comum. Tal como os critérios anteriores, há a validação dos anúncios, seguindo-se, caso seja necessário, aplicação de critérios de desempate enunciados anteriormente (Figura 3.11).

No final destas quatro fases do algoritmo, espera-se que um anúncio válido seja devolvido.

3.4 Sumário

Neste capítulo foram abordadas as estruturas dos programas e dos anúncios, os metadados que os constituem e as regras por detrás da sua criação. Os programas são simplesmente constituídos por um formato e conjunto de géneros, enquanto que os anúncios têm uma estrutura muito mais complexa, devido ao conjunto de critérios que os anunciantes lhes atribuem.

Ainda é apresentada uma metodologia a ser aplicada para resolução do problema, isto é, como o algoritmo de classificação e seleção de conteúdo irá processar os dados relativos dos programas e anúncios. Foram distinguidas duas fases principais, a do pré-processamento e a do *matching*, dividindo-se esta última em duas partes, uma *offline* e outra *online* (Figura 3.12), devido ao facto de lidar com diferentes tipos de atributos (estáticos e dinâmicos).

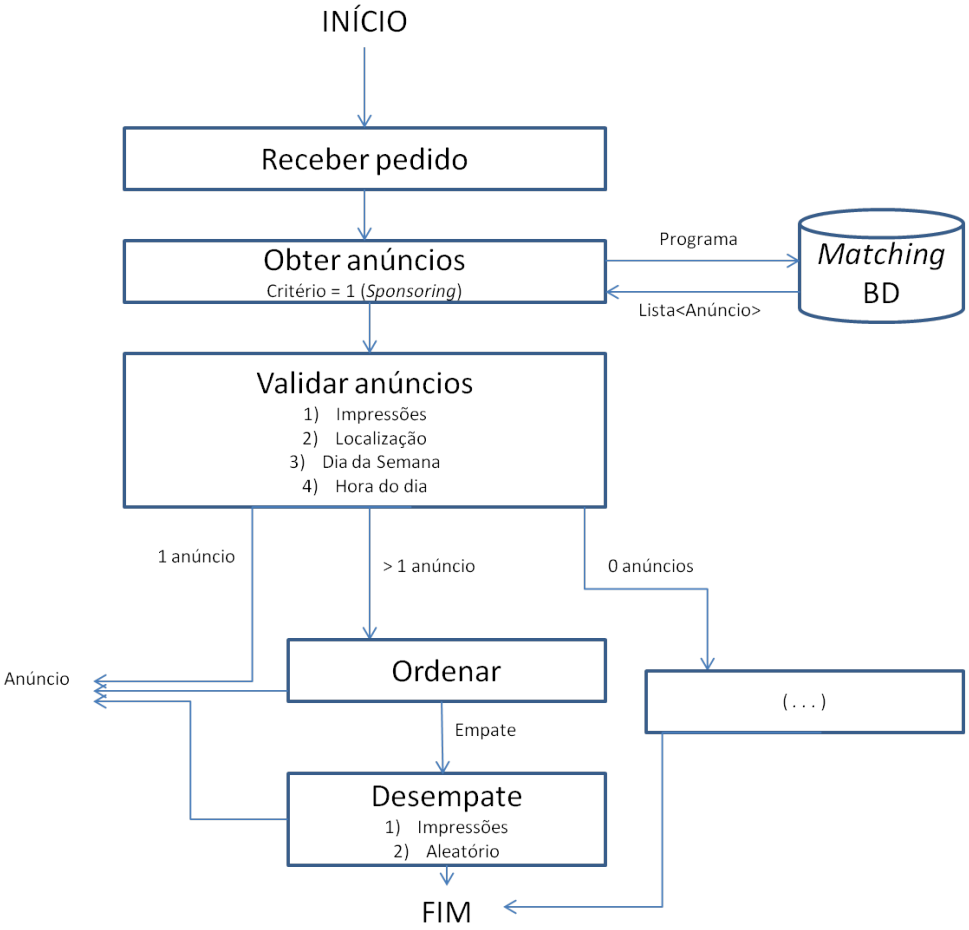

Figura 3.8: *Matching Online* por *sponsoring*

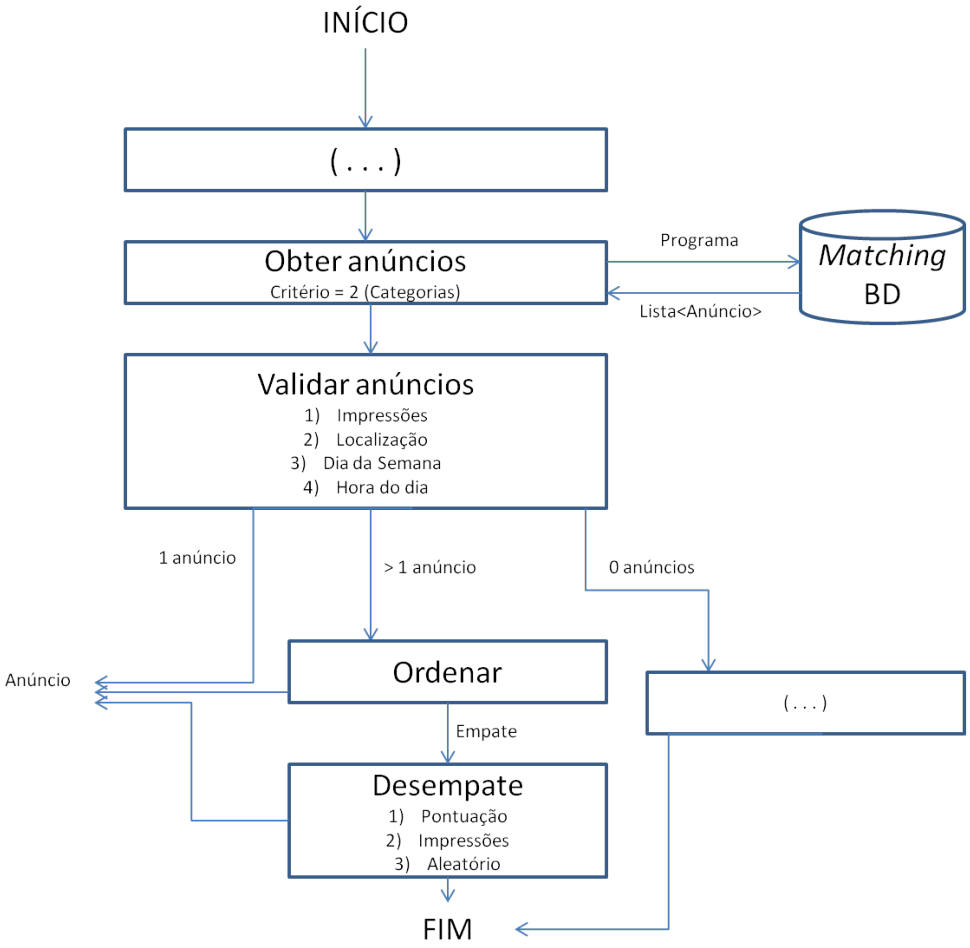

Figura 3.9: *Matching Online* por categorias

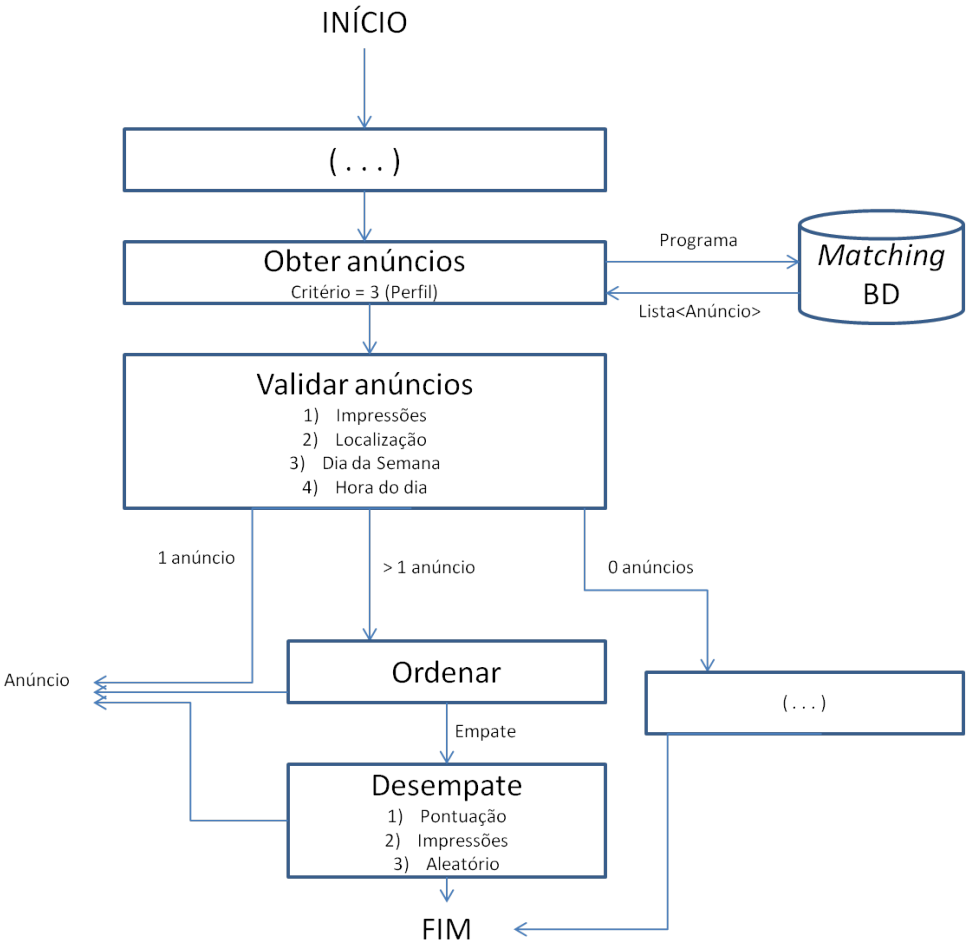

Figura 3.10: *Matching Online* por perfil

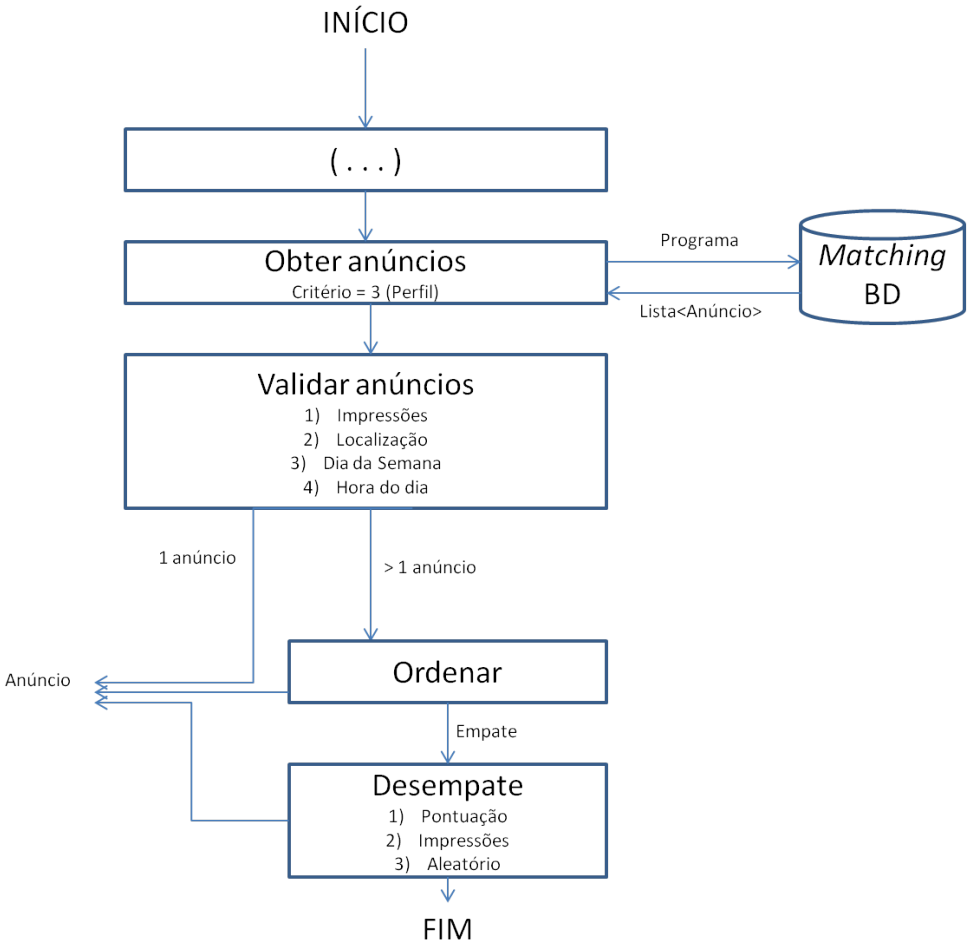

Figura 3.11: *Matching Online* por regras

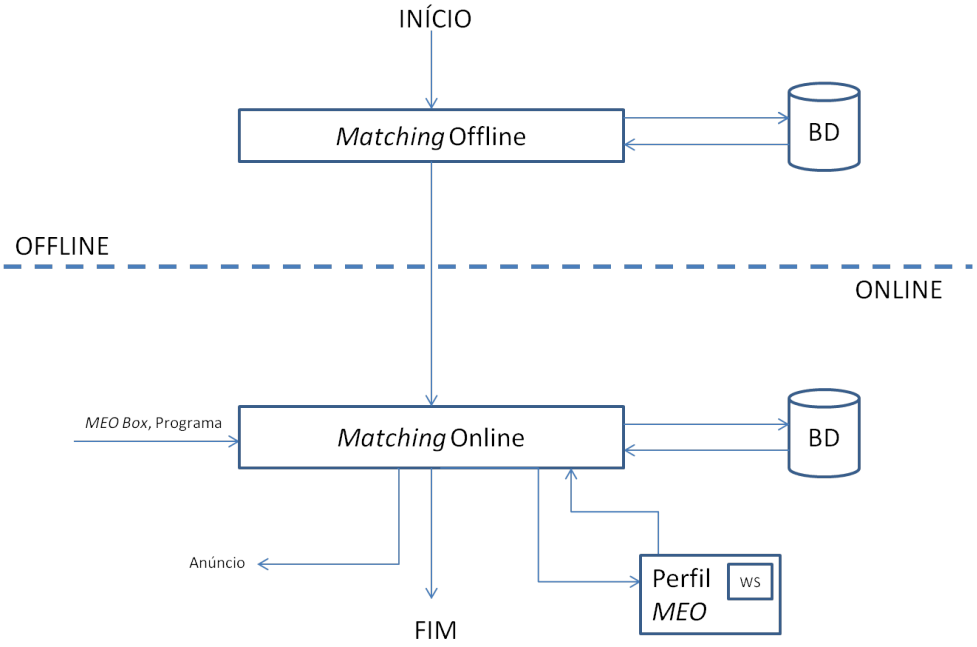

Figura 3.12: Visão geral do *matching*

Capítulo 4

Implementação

A análise do problema levou a que uma metodologia fosse desenvolvida para a sua resolução. Aplicação dessa metodologia depende que todos os componentes do sistema estejam no local certo e com todas as funcionalidades bem definidas, assim como o local onde serão armazenados todos os dados e resultados relevantes.

Neste capítulo é feita uma análise às arquiteturas física e lógica do sistema, sendo ainda feita uma decomposição horizontal e vertical desta última. Aqui também é descrita a base de dados relacional projetada para o armazenamento de todos os dados e gestão do sistema. Conclui-se o capítulo com uma análise ao esforço despendido na realização das tarefas estabelecidas.

4.1 Arquitetura do Sistema

Todo o sistema é composto por uma arquitetura simples, do tipo cliente-servidor, mas ainda com alguns componentes externos, como o serviço que trata dos anúncios e o que fornece as informações de perfil de uma *MEO Box*.

4.1.1 Arquitetura Física

Na figura 4.1 é apresentado um esquema da arquitetura física geral de todo o sistema. Como foi referido anteriormente, o algoritmo de *matching* está integrado num serviço (*Algorithm Server*) que responde a pedidos de anúncios por parte de outro serviço exterior, um serviço-cliente (*MEO Service*). Este último está ligado diretamente à *MEO Box*, de onde recebe a informação que o utilizador está a visualizar um determinado conteúdo televisivo. O serviço que processa o pedido, comunica com uma base de dados que está localizada na mesma máquina, que contém toda a informação dos programas, anúncios e respetivos pares, assim como recorre a outros serviços, como o que dá as informações de perfil para uma determinada *MEO Box*.

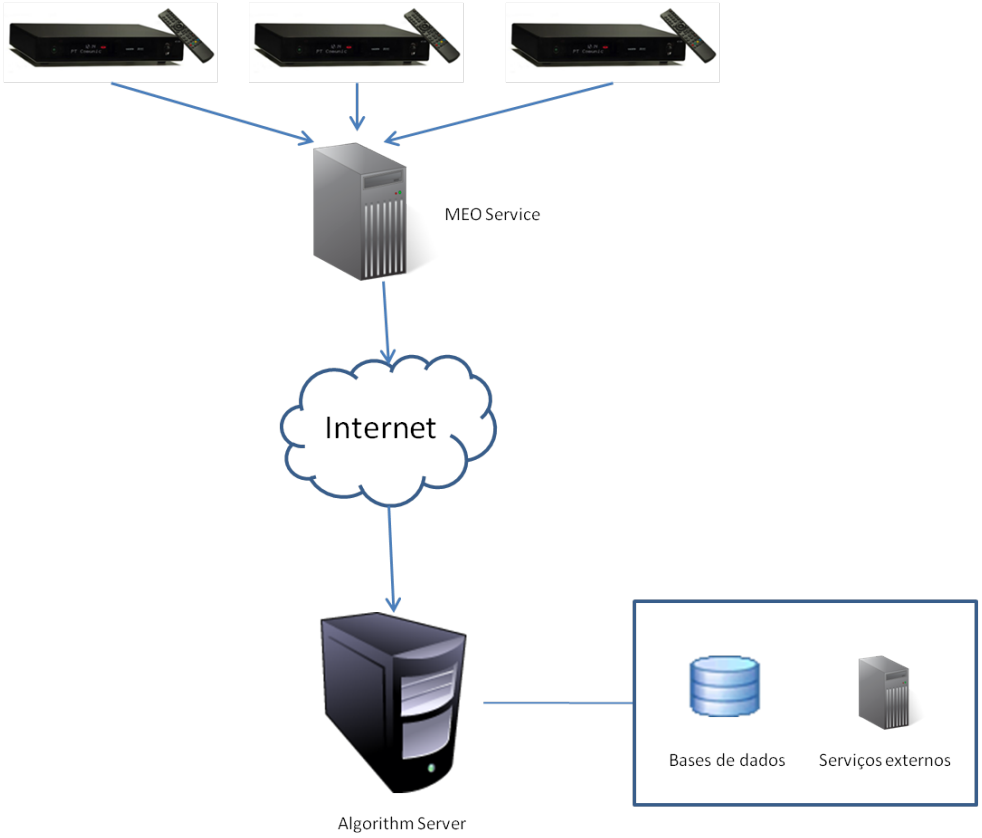

Figura 4.1: Arquitetura Física

4.1.2 Arquitetura Lógica

O sistema pode ser decomposto também descrito através de uma decomposição horizontal e, de uma forma mais detalhada, por uma decomposição vertical. A decomposição lógica horizontal apresenta a estrutura lógica do sistema de forma relativamente independente das funcionalidades deste, isto é, não referindo componentes específicos. Por sua vez, na decomposição lógica vertical são apresentadas as diferentes funcionalidades do sistema agrupando-as em subsistemas.

4.1.2.1 Arquitetura Lógica Horizontal

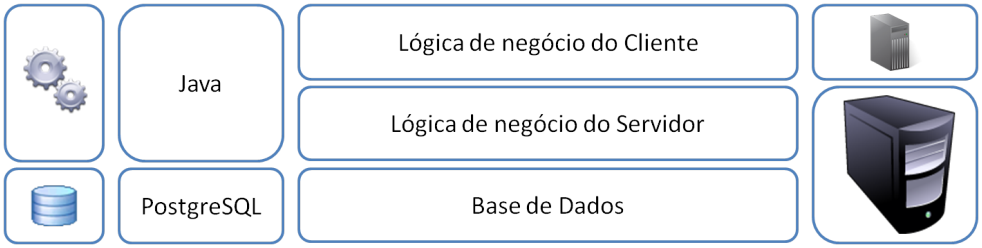

Figura 4.2: Arquitetura Lógica Horizontal

A figura 4.2 demonstra como está dividido o sistema por camadas distintas. Não existe uma interface gráfica de utilizador visto não ser necessário para o âmbito deste projeto, logo a primeira camada é a lógica de negócio do cliente, que trata de receber os pedidos do lado de fora do serviço-cliente, enviando-os para o servidor, receber os resultados e devolvê-los ao serviço cliente. A lógica de negócio do servidor está encarregue de processar os pedidos, fazendo o *matching online* (tempo real), e ainda tratar do pré-processamento dos dados, que envolve gerir programas e anúncios e respetivos emparelhamentos (*matching offline*), e comunicar com serviços exteriores quando há necessidade de recorrer a informações de perfil. A última camada, a camada dos dados, é mantida através de uma base de dados *PostgreSQL*, permitindo a gestão de todos os anúncios, programas e relações existentes.

4.1.2.2 Arquitetura Lógica Vertical

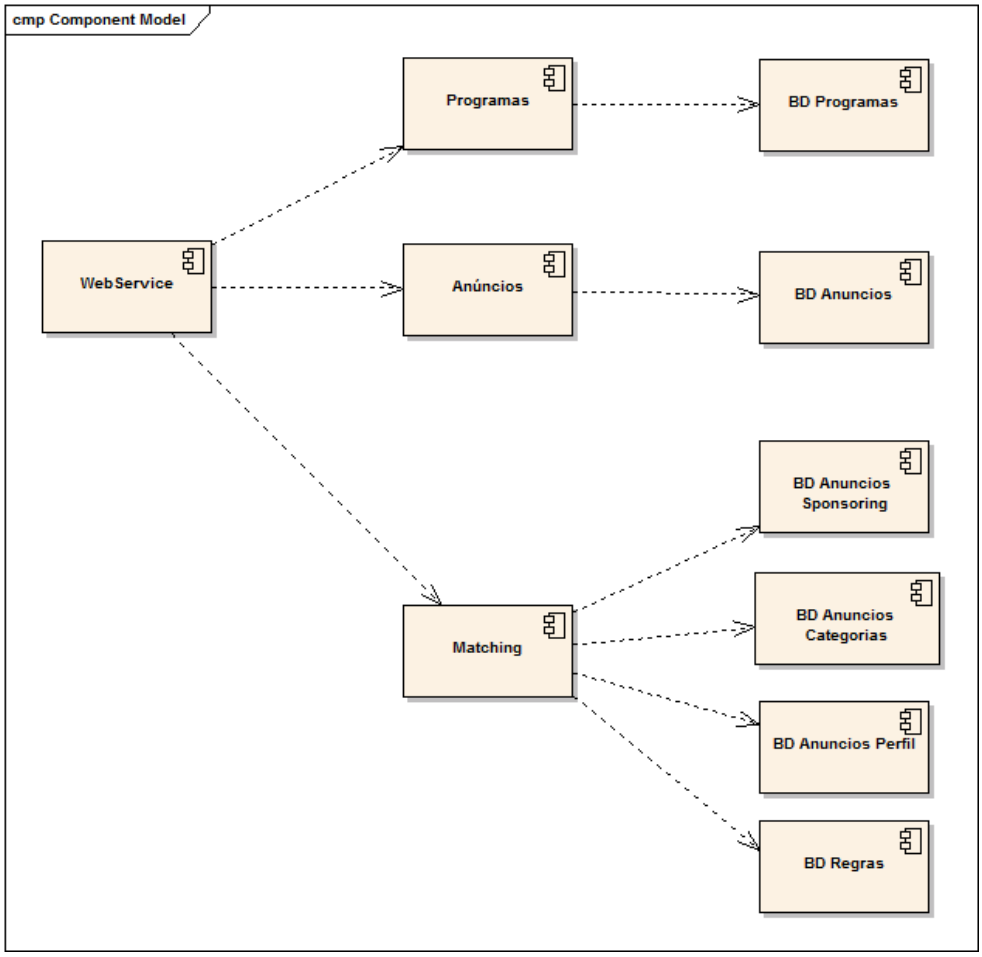

Figura 4.3: Arquitetura Lógica Vertical

No nível mais acima encontra-se o módulo do *WebService* através do qual o cliente interage, fazendo a gestão de pedidos. Permite visualizar o conjunto de metadados dos anúncios e programas, assim como pedir um anúncio para um determinado programa, numa determinada *MEO*

Box.

O módulo seguinte é o dos Programas que comunica com a base de dados e permite devolver conjuntos de metadados. Comunica ainda gerir os componentes intervenientes, garantindo que apenas programas válidos estejam armazenados. O módulo dos Anúncios é semelhante ao dos Programas, pois trata de guardar e gerir todos os metadados de anúncios. Este módulo está responsável também pela visualização de anúncios e ainda atualizações feitas a estes.

O módulo *Matching* é o módulo principal do sistema, pois gere todas as relações entre programas e anúncios ao nível de *sponsoring*, categorias e perfil, trata da extração e atualização das regras de associação que possam existir assim como é responsável pelo *Matching* por este último critério. É o módulo responsável por satisfazer pedidos de anúncios aquando da visualização de um programa.

O acesso aos dados é feito recorrendo a uma API escrita em Java designada por JDBC (*Java Database Connectivity*) que permite enviar instruções SQL a bases de dados relacionais.

4.2 Base de dados

Como foi referido anteriormente, os dados são mantidos usando uma base de dados em *PostgreSQL*. Existe uma gestão contínua conforme foi referido no capítulo 3, que permite que apenas sejam armazenados dados relevantes para a resolução dos pedidos que o serviço irá receber. A figura 4.4, juntamente com a tabela 4.1, descrevem a base de dados relacional usada neste sistema, e respetiva normalização.

As tabelas referentes aos programas são: programsimple, format, genre e programsimplegenre. Os anúncios são mantidos pelo conjunto de relações entre as tabelas: ad, macro, value, advalue, timeinterval, adtimeinterval, location adlocation, locationpostalcode, targetsponsoring, targetformatgenre, targetprogram, targetprofile. Quanto as resultados do matching, esses são guardados na tabela match.

4.3 Esforço de desenvolvimento

O desenvolvimento do sistema iniciou-se a 15 de Fevereiro de 2012, tendo data de previsão para terminar no dia 15 de Julho, perfazendo um total de 21 semanas. Ao longo destas semanas foram, desenvolvidos os diferentes componentes que constituem o sistema.

O trabalho foi dividido em 4 fases principais:

1. Análise do Problema - Para esta fase, fora dispensadas 4 semanas, envolvendo toda a parte teórica antes de começar a implementação do sistema em si. Esta fase, foi dividida em 3 iterações diferentes:

    - Esquema conceptual do sistema;
    - Funcionalidades do serviço;

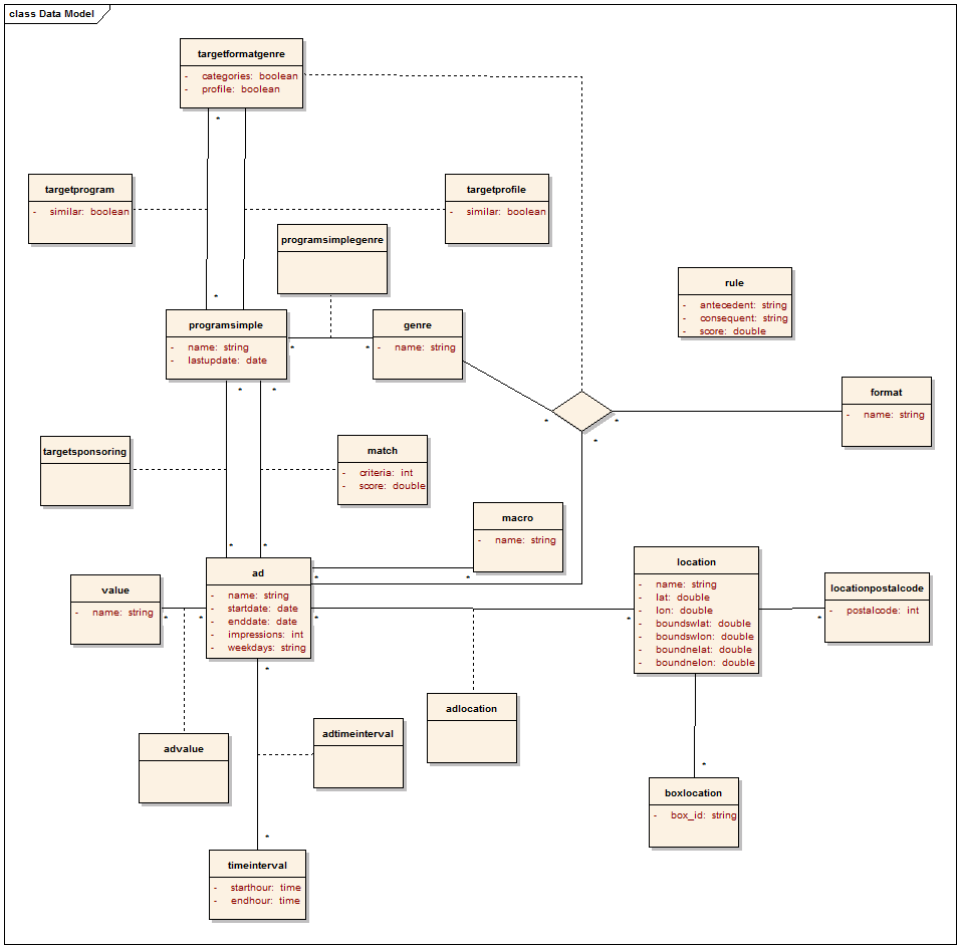

Figura 4.4: Esquema da base de dados relacional

- Desenho da base de dados;

2. Desenvolvimento e Integração - Esta fase durou 10 semanas, sendo a fase que mais tempo tomou. Os componentes aqui desenvolvidos são aqueles que irão fazer com que o sistema funcione e cumpra com os requisitos. Esta fase, foi dividida em 7 iterações diferentes:

   - Matching Offline - Critério 1;
   - Matching Offline - Critério 2 e 3;
   - Extração de regras;
   - Matching Offline - Critério 4;
   - Cálculo de semelhança entre programas e anúncios;
   - Matching Online
   - Integração com o serviço;

3. Validação - Esta fase de validação, compreende grande parte do período de desenvolvimento e ainda mais 2 semanas, devido ao facto de cada componente desenvolvido durante

Tabela	Atributos	Descrição
programsimple	*id*, name, lastupdate	Programas
format	*id*, name	Formatos
genre	*id*, name	Géneros
programsimplegenre	*programsimple_id* → *programsimple*, *genre_id* → *genre*	Géneros dos programas
ad	*id*, name, stardate, enddate, impressions, weekdays	Géneros
macro	*id*, name	Géneros
value	*id*, name	Géneros
advalue	*ad_id* → *ad*, *value_id* → *value*	Valores dos anúncios
timeinterval	*id*, starhour, endhour	Intervalos de tempo
adtimeinterval	*ad_id* → *ad*, *timeinterval_id* → *timeinterval*	Intervalos de tempo dos anúncios
location	*id*, name, lat, lon, boundswlat, boundswlon, boundnelat, boundnelon	Cidades
adlocation	*ad_id* → *ad*, *location_id* → *location*	Cidades dos anúncios
boxlocation	*box_id*, *location_id* → *location*	Cidades das MEO Boxes
locationpostalcode	*location_id* → *location*, *postalcode*,	Códigos postais das cidades
targetsponsoring	*ad_id* → *ad*, *programsimple_id* → *programsimple*	Sponsoring dos anúncios
targetformatgenre	*ad_id* → *ad*, *format_id* → *format*, *genre_id* → *genre*	Formatos/Géneros dos anúncios
targetprogram	*ad_id* → *ad*, *format_id* → *format*, *genre_id* → *genre*, *programsimple_id* → *programsimple*, similar	Categorias (público-alvo) dos anúncios
targetprofile	*ad_id* → *ad*, *format_id* → *format*, *genre_id* → *genre*, *programsimple_id* → *programsimple*, similar	Perfil (público-alvo) dos anúncios
match	*ad_id* → *ad*, *programsimple_id* → *programsimple*, criteria, score	Pares programa-anúncio
rule	*antecedent*, *consequent*, score	Regras de associação

Tabela 4.1: Base de dados normalizada

a segunda fase ter sido devidamente validado, individualmente, e depois integrado com os restantes já desenvolvidos.

4. Escrita do relatório - As restantes 5 semanas foram dedicadas à escrita deste relatório, assim como para fazer pequenos ajustes e modificações.

4.4 Sumário

Neste capítulo foi apresentada a arquitetura do sistema desenvolvido, a diferentes níveis. A arquitetura física mostra como os componentes físicos do sistema estão distribuídos. A arquitetura lógica, apresenta o sistema mais a um nível funcional, em que a arquitetura lógica horizontal mostra o sistema dividido por camadas enquanto que a arquitetura lógica vertical, apresenta o sistema mais a nível de componentes existentes.

É também ainda apresentada o esquema relacional da base de dados usada pelo sistema, capaz de manter e gerir todos os dados referentes a programas, anúncios e pares programas-anúncios que são criados e armazenados.

Para finalizar, foram apresentadas as diferentes iterações ao longo das 21 semanas deste projeto.

Capítulo 5

Avaliação

A avaliação de sistemas é uma componente bastante importante do seu desenvolvimento de forma a avaliar a eficácia e desempenho do mesmo. Foram realizados testes a diferentes níveis do sistema, de forma a avaliar o seu funcionamento. Os testes foram realizados sobre uma base de dados numa máquina remota, usando um túnel SSH (*Secure Shell*) para a comunicação direta com esta, a partir de uma máquina externa.

Tanto o *webservice* como os algoritmos desenvolvidos foram implementados usando como linguagem de programação Java e como ambiente de desenvolvimento *NetBeans*. A base de dados *PostgreSQL* encontrava-se localizada numa máquina remota, em ambiente Linux.

Neste capítulo são apresentados alguns resultados de testes realizados, assim como uma comparação qualitativa entre o serviço desenvolvido e os serviços existentes referidos na revisão bibliográfica.

5.1 Pré-processamento

Esta primeira fase envolve, conforme referido anteriormente, o carregamento diário de programas e de anúncios, ou seja, operações como adicionar, atualizar e apagar instâncias destes.

Quanto ao programas, é percorrida toda a lista de instâncias para um determinado dia, podendo haver novos programas,o que leva a adicionar estes, assim como atualizar a data de validade de programas já existentes, e apagar aqueles que já não serão válidos, pelo critério das duas semanas de validade desde a sua última atualização.

Esta fase do pré-processamento envolve uma complexidade temporal de O(n) no melhor dos casos e de O(nxm) no pior deles, em que n representa o número de programas existentes na base de dados e m representa o número de programas da lista diária.

Quanto aos anúncios, trata-se de um processo mais complexo, pois estes envolvem muitas mais tabelas e relações a gerir. A tabela 5.1 apresenta os resultados de testes feitos para testar o tempo que demora ao sistema carregar diferentes conjuntos de anúncios.

	Número de anúncios			
	1	10	100	1.000
Tempo (*ms*) para carregar	700	6.137	71.684	633.359

Tabela 5.1: Tempos de carregamento de anúncios

Neste teste, cada anúncio criado, continha uma macro, três valores, um programa associado ao critério do *sponsoring*, dois ao critérios das categorias e um ao critério do perfil, assim como informações relativos à campanha (data de início, data de fim, número de impressões, dia da semana, horas do dia e uma localização).

O conjunto de anúncios foi gerado automaticamente, de forma aleatória, por forma a testar os três tipos de situações com que o serviço pode vir a enfrentar:

- 1 - Durante o dia, podem ser criados novos anúncios e as notificações que chegam ao serviço fazem-se acompanhar dos metadados do novo anúncio;
- 10 e 100 - Ao fazer *pooling* diário ao serviço de anúncios, podem ter sido criados anúncios, dos quais não houve notificação, sendo esses carregados do serviço em quantidades entre 10 e 100;
- 1000 - Ao reiniciar o sistema e/ou as bases de dados, foi preciso saber o tempo que o sistema demora a adicionar uma grande quantidade de anúncios fora do normal.

5.2 *Matching*

A objetivo principal do *webservice* desenvolvido, integrado com os algoritmos de extração de regras e seleção de conteúdos, é conseguir que devolva um anúncio válido. Para isso, é necessário que anúncios estejam atribuídos a programas, o que envolve algum processamento por parte do serviço. Daí que o processo de *matching* tenha sido dividido em duas fases distintas, que lidam com critérios diferentes, o *matching offline* e *matching online*.

5.2.1 *Matching Offline*

O *matching offline* lida com os critérios estáticos, ou seja, apenas aqueles que não se alteram ou que se alteram com pouca frequência. Estes critérios podem ser aplicados a pares programa-anúncio *apriori*, ou seja, ainda sem qualquer pedido por um anúncio tenha sido feito. A tabela 5.2 demonstra os resultados obtidos nos testes de performance executados, em termos de tempos de realização das tarefas de *matching* pelos critérios de *sponsoring*, categorias e perfil.

Como seria de esperar, os tempos de *matching* pelo critério de *sponsoring* são inferiores aos dos restantes critérios pelo simples facto de não haver qualquer cálculo de pontuação, entre cada par programa-anúncio e, consequente, trata-se de uma operação relativamente rápida. Quanto aos restantes critérios, o conjunto de dados usados foi o mesmo do primeiro teste (Tabela 5.1), daí que os tempos de *matching* pelo critério do perfil sejam relativamente inferiores, em cerca de

	Número de anúncios		
	1	10	100
Sponsoring	400	702	1.102
Categorias	556	4.073	19.788
Perfil	428	2.280	9.781

Tabela 5.2: Tempos de *matching offline*: sponsoring, categorias e perfil

metade, que *matching* pelo critério das categorias (o critério das categorias tinha dois programas associados e o de perfil apenas um).

O *matching offline* compreende ainda uma fase de extração de associações para a criação das regras, o nosso quarto critério de *matching*. Para uma análise da eficiência comparativa entre possíveis métodos a serem usados, foram feitos testes ao algoritmo usado para extração de regras, sem suporte mínimo, ou seja, sem qualquer valor mínimo de frequência relativa, e com suporte mínimo de 5%, ou seja, no conjunto de todas as associações extraídas, as únicas consideradas relevantes terão que ter uma frequência relativa de 5%.

		Número de pares programa-anúncio		
		1	10	100
Sem suporte mínimo	Tempo (*ms*)	847	2.500	20.444
	Nº de regras	4	151	3.053
Com suporte mínimo (5%)	Tempo (*ms*)	392	1.289	10.360
	Nº de regras	4	15	93

Tabela 5.3: Extração de regras

Os resultados obtidos em 5.3 demonstram que havendo um controlo mínimo sobre o conjunto de regras obtido faz com que o tempo de extração destas seja, consequentemente, inferior.

Um incremento do número de pares existentes faz com que o conjunto de regras a extrair possa crescer exponencialmente, daí que é recomendado o uso de suporte mínimo para um melhor desempenho, tanto em termos de tempo como de qualidade, pois o suporte mínimo para além de rapidez garante que as regras extraídas sejam relevantes no conjunto. O conjunto de pares programa-anúncio usado para este teste foi gerado automaticamente, de forma aleatória.

O número de regras extraídas e construídas é de extrema importância pois influencia o processo de *matching* pelas mesmas. Nesta fase final, o número de anúncios a processar será em muito superior ao total dos critérios anteriores. O teste seguinte tem como objetivo determinar como o número de pares programa-anúncio, criados pelos critérios de *sponsoring*, categorias e perfil, influencia o número de pares gerados pelas regras extraídas destes, assim como o seu desempenho em termos de tempo. Para este teste, foram gerados conjuntos de pares programa-anúncio de forma aleatória.

Mais uma vez, os resultados obtidos (Tabela 5.4) demonstram que sem haver um controlo sobre o número de regras a ser gerado, os resultados obtidos irão ser superiores por uma grande margem. Também, tal como se previa, o número de pares gerados é muito superior ao total de

		Número de pares programa-anúncio (Critérios 1, 2, 3)		
		1	10	100
Sem suporte mínimo	Nº de regras	31	508	4.312
	Nº de pares gerados	3.976	172.482	1.021.379
	Tempo (*ms*)	6.120	121.666	695.114
Com suporte mínimo (5%)	Nº de regras	31	391	1.359
	Nº de pares gerados	3.976	34.132	99.603
	Tempo (*ms*)	6.088	76.679	268.929

Tabela 5.4: *Matching* pelo critério 4

pares presentes no conjunto dos outros três critérios, aumentando assim a probabilidade do último critério conseguir providenciar um anúncio adequado.

5.2.2 *Matching Online*

O *matching online* consiste na seleção de um anúncio em tempo real, baseando-se em critérios dinâmicos. O processo de *matching online* necessita do identificador da *MEO Box* de onde vem o pedido, e do programa que está a ser visualizado de forma a poder proceder à seleção do anúncio. A seleção, tal como foi referido anteriormente, consistem em percorrer os pares programa-anúncios construídos no *matching offline*, separados por critérios, validando cada um dos grupos, até que um anúncio válido seja selecionado. Por forma a testar o desempenho do sistema neste ponto, foram realizados uma série de pedidos sequenciais ao serviço, sabendo que seria sempre devolvido um resultado. A tabela seguinte demonstra os resultados obtidos.

	Número de pedidos			
	1	10	100	1.000
Tempo (*ms*) de resposta	463	2.277	26.820	279.966

Tabela 5.5: Tempos de resposta a pedidos de anúncio

5.3 Avaliação Qualitativa

No capítulo 2 foram avaliados dois serviços de distribuição de publicidade contextualizada. Estes serviços, à semelhança do que foi desenvolvido apresenta bastantes pontos em comum, nomeadamente o facto de permitir aos anunciantes escolher os critérios que irão definir o público-alvo em questão.

Espacial e temporal Tanto o *Advertising.com* e o *Clicksor.com* como o serviço que foi desenvolvido, permitem ao anunciante definir os locais (posição) e horas do dia em que o anúncio poderá ser exibido, visto este ser um critério de maior importância, pois interessa ao anunciante que as pessoas vejam os seus anúncios durante um horário que se saiba que tem público.

Redirecionamento Ambos os serviços analisados aproveitam dados históricos de consultas feitas anteriormente na Internet a outros produtos, para distribuição de publicidades que lhes possam interessar. O serviço aqui desenvolvido utiliza informações semelhantes, no que diz respeito a programas vistos anteriormente, isto é, um dos critérios que o anunciante pode escolher é o do perfil, que se baseia nos programas preferidos dos utilizadores, podendo considerar-se equiparável a dados históricos.

Audiências modeladas Apesar do critério anterior poder relacionar-se com o do perfil de um cliente, este critério da *Advertising.com* ainda mais semelhanças tem, pois baseia-se nos pontos em comum que cada cliente terá com públicos-alvo definidos para anúncios.

Palavras-chave Este critério da *Clicksor.com* pode ser comparado com o critério das categorias do serviço desenvolvido, pois são escolhidos formatos e/ou programas servindo como base para o *matching* por este critério.

Quanto aos restantes critérios, alguns são comuns entre os dois serviços como o caso do Tecnográfico (*Advertising.com*) e dos Dispositivos e Sistema Operativo (*Clicksor.com*), mas nada em comum têm com os critérios do serviço desenvolvido. Isto deve-se também ao facto que a *Advertising.com* e a *Clicksor.com* distribuem publicidade pela Internet, enquanto que o serviço desenvolvido no âmbito deste projeto foi apenas conceptualizado para o serviço da *MEO*.

5.4 Sumário

Neste capítulo foram realizados vários testes com o objetivo de testar e esforçar o sistema, avaliando o seu desempenho e performance, tanto no que toca a tempos de resolução, assim como qualidade dos dados.

A extração de regras para a construção mostrou resultados satisfatórios, no que toca ao número de anúncios que poderiam ser emparelhados com programas, e ainda permitiu concluir que aumentando o suporte mínimo para criação de regras, o número de regras decresce e diminui-se substancialmente o número de pares gerados pelo *matching* por este critério, aumentando ao mesmo tempo a qualidade destas, pois a regra só existirá se realmente se revelar uma tendência, que é o objetivo principal deste processo.

Foi ainda feita uma análise comparativa entre os serviços analisados na revisão bibliográfica e o serviço desenvolvido. Existem pontos em comum, pelo simples facto de se tratar de distribuição de publicidade contextualizada, mas também o facto de terem ambientes de ação diferentes, leva a que tenham bastante diferenças.

Capítulo 6

Conclusões e Trabalho Futuro

O principal objetivo deste projeto era o desenvolvimento de um serviço capaz de recorrer a serviços externos e de classificar anúncios com base num conjunto de critérios definidos pelo anunciante. Este objetivo principal foi cumprido e ainda foi implementado um algoritmo de extração de regras de associação, baseado no *H-Mine*, que demonstrou ser bastante eficaz no que toca a emparelhar programas com anúncios, podendo trocar quantidade por qualidade apenas alterando o valor de suporte mínimo, único critério existente para que uma determinada regra possa ser criada.

O algoritmo funciona, em coordenação com a base de dados, como um algoritmo de aprendizagem, em que por cada novo par programa-anúncio, ele aprende os conjuntos formato/género e macros/valores associados, podendo usar essa informação para o processo de *matching* por regras.

A principal dificuldade encontrada foi relativamente à metodologia a implementar, devido ao facto da existência de critérios dinâmicos e estáticos. Optou-se por uma solução que conseguisse "prever"pares programa-anúncio, apenas com base nos critérios estáticos, deixando apenas de fora, para um *matching online*, aqueles que não fossem possíveis de prever, os dinâmicos.

O planeamento e identificação das iterações ajudou a que o processo de desenvolvimento fosse fluido e consistente.

Concluindo, este trabalho era algo bastante ambicioso, por aliar tantos componentes externos, tantos critérios para seleção de anúncios, uma aprendizagem automática com uma arquitetura funcional, tanto a nível *online* como *offline*. No entanto os objetivos principais foram cumpridos, sendo possível usar o serviço para distribuir publicidade baseada em informação de contexto.

6.1 Trabalho Futuro

Quanto à extração de regras, esta pode ser integrada com a componente do serviço externo que lida com a criação de anúncios. Assim, é possível dar ao anunciante a possibilidade de saber quais as "tendências"atuais, isto é, pedir uma "recomendação"do público-alvo para o anúncio que acabou de criar, agilizando ao mesmo tempo o processo de criação de anúncios.

O processo de seleção de anúncios pode ser reforçado recorrendo a técnicas de cálculo de programas semelhantes, introduzindo este cálculo como um novo critério, caso o das regras falhasse. Ao selecionar o público-alvo (categorias, perfil), poderia selecionar programas semelhantes aos que selecionou, entrando esses programas como um novo critério, ou seja, todo o processo de *matching online* pelos critérios de *sponsoring*, categorias, perfil e regras seria aplicado a programas semelhantes ao que o utilizador está a consumir, caso este não tenha qualquer anúncio válido para o contexto do utilizador, garantindo assim um maior leque de opções. Resultante disto, poderia ser o aumento do suporte mínimo para extração das regras, aumentando a qualidade destas e garantindo que a procura resultaria num "bom anúncio".

Quanto ao serviço em si, podia ser usado também para serviços de televisão móvel, nomeadamente o *MEO GO!* [kn:12d], podendo aproveitar todo o sistema desenvolvido.

Referências

[AAP00] Ramesh C. Agarwal, Charu C. Aggarwal e V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. *Journal of Parallel and Distributed Computing*, 61:350–371, 2000.

[AS94] Rakesh Agrawal e Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[AT05] G. Adomavicius e A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734 – 749, june 2005.

[Ham50] R W Hamming. *Bell System Technical Journal*, 29(2):147–160, 1950.

[HKP06] Jiawei Han, Micheline Kamber e Jian Pei. *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 2 edition, January 2006.

[HPY00] Jiawei Han, Jian Pei e Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29:1–12, May 2000.

[Jac01] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[kn:12a] Advertising.com, 2012. Disponível em http://www.advertising.com/, acedido em Junho, 2012.

[kn:12b] Clicksor.com, 2012. Disponível em http://www.clicksor.com/, acedido em Junho, 2012.

[kn:12c] Meo, 2012. Disponível em http://www.meo.pt, acedido em Junho, 2012.

[kn:12d] Meo go!, 2012. Disponível em http://www.meo.pt/ver/meogo, acedido em Junho, 2012.

[kn:12e] Pt inovação, 2012. Disponível em http://www.ptinovacao.pt/, acedido em Junho, 2012.

[kn:12f] Sapo, 2012. Disponível em http://www.sapo.pt, acedido em Junho, 2012.

[kn:12g] Slides da cadeira de extracção de conhecimento e aprendizagem computacional da faculdade de engenharia da universidade do porto, 2012. Disponível em http://

paginas.fe.up.pt/~ec/index.html/, acedido em Dezembro, 2011 e Janeiro, 2012.

[PHL+01] Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang e Dongqing Yang. H-mine: Hyper-structure mining of frequent patterns in large databases. In *ICDM'01*, pages 441–448, 2001.

[PRB+08] Nearchos Paspallis, Romain Rouvoy, Paolo Barone, George A. Papadopoulos, Frank Eliassen e Alessandro Mamelli. A pluggable and reconfigurable architecture for a context-aware enabling middleware system. In *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems:*, OTM '08, pages 553–570, Berlin, Heidelberg, 2008. Springer-Verlag.